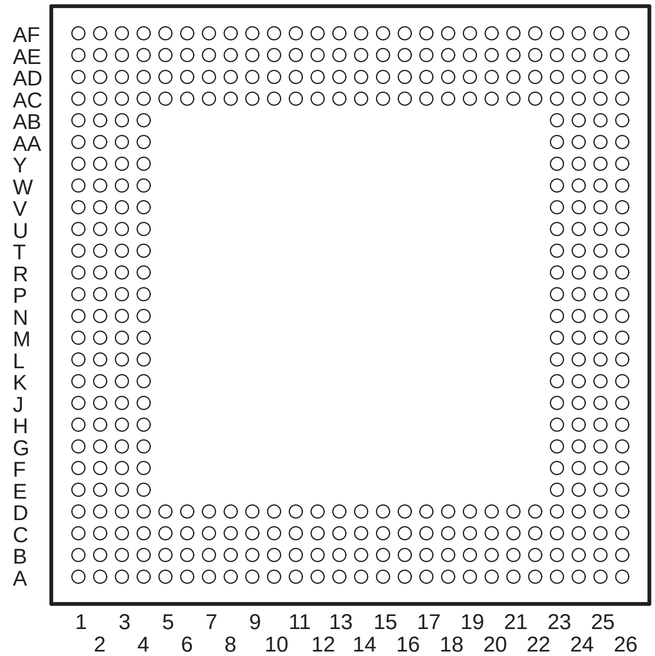
- **Highest Performance Fixed-Point Digital Signal Processor (DSP) TMS320C6202**
  - **4-ns Instruction Cycle Time**
  - **250-MHz Clock Rate**
  - **Eight 32-Bit Instructions/Cycle**
  - **2000 MIPS**
- **VelociTI™ Advanced Very Long Instruction Word (VLIW) ’C6200 CPU Core**
  - **Eight Highly Independent Functional Units:**
    - **Six ALUs (32-/40-Bit)**
    - **Two 16-Bit Multipliers (32-Bit Result)**
  - **Load-Store Architecture With 32 32-Bit General-Purpose Registers**
  - **Instruction Packing Reduces Code Size**
  - **All Instructions Conditional**
- **Instruction Set Features**
  - **Byte-Addressable (8-, 16-, 32-Bit Data)**
  - **32-Bit Address Range**
  - **8-Bit Overflow Protection**
  - **Saturation**
  - **Bit-Field Extract, Set, Clear**
  - **Bit-Counting**
  - **Normalization**
- **3M-Bit On-Chip SRAM**
  - **2M-Bit Internal Program/Cache**
    - **Two 128K-Byte Blocks Offer Improved Concurrency**
      **Block 0: 128K Bytes Memory-Mapped**
      **Block 1: 128K Bytes Direct-Mapped Cache/Memory-Mapped**
  - **1M-Bit Dual-Access Internal Data (128K Bytes)**
    - **Two 64K-Byte Blocks Offer Improved Concurrency**
- **32-Bit External Memory Interface (EMIF)**
  - **Glueless Interface to Synchronous Memories: SDRAM or SBSRAM**
  - **Glueless Interface to Asynchronous Memories: SRAM and EPROM**
- **Four-Channel Bootloading Direct-Memory-Access (DMA) Controller With an Auxiliary Channel**
- **Flexible Phase-Locked-Loop (PLL) Clock Generator**
- **32-Bit Expansion Bus**
  - **Glueless/Low-Glue Interface to Popular PCI Bridge Chips**
  - **Glueless/Low-Glue Interface to Popular Synchronous or Asynchronous Microprocessor Buses**
  - **Master/Slave Functionality**
  - **Glueless Interface to Synchronous FIFOs and Asynchronous Peripherals**
- **Three Multichannel Buffered Serial Ports (McBSPs)**
  - **Direct Interface to T1/E1, MVIP, SCSA Framers**
  - **ST-Bus-Switching Compatible**
  - **Up to 256 Channels Each**
  - **AC97-Compatible**
  - **Serial-Peripheral-Interface (SPI) Compatible (Motorola™)**
- **Two 32-Bit General-Purpose Timers**
- **IEEE-1149.1 (JTAG†) Boundary-Scan-Compatible**
- **352-Pin BGA Package (GJL Suffix)**
- **384-Pin BGA Package (GLS Suffix)**
- **0.18-μm/5-Level Metal Process**
  - **CMOS Technology**
- **3.3-V I/Os, 1.8-V Internal**

**ADVANCE INFORMATION**

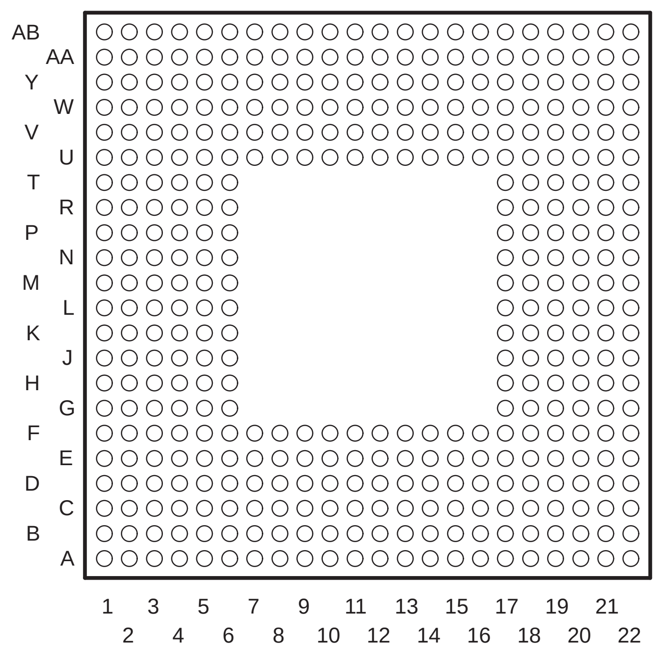

† IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture.

ADVANCE INFORMATION

**GJL 352-PIN BALL GRID ARRAY (BGA) PACKAGE**
**( BOTTOM VIEW )**




**GLS 384-PIN BALL GRID ARRAY (BGA) PACKAGE**
**( BOTTOM VIEW )**



## description

The TMS320C62x DSPs (including the TMS320C6202 device) are the fixed-point DSP family in the TMS320C6000 platform. The TMS320C6202 (’C6202) device is based on the high-performance, advanced VelociTI very-long-instruction-word (VLIW) architecture developed by Texas Instruments (TI™), making this DSP an excellent choice for multichannel and multifunction applications.

With performance of up to 2000 million instructions per second (MIPS) at a clock rate of 250 MHz, the ’C6202 offers cost-effective solutions to high-performance DSP programming challenges. The ’C6202 DSP possesses the operational flexibility of high-speed controllers and the numerical capability of array processors. This processor has 32 general-purpose registers of 32–bit word length and eight highly independent functional units. The eight functional units provide six arithmetic logic units (ALUs) for a high degree of parallelism and two 16-bit multipliers for a 32-bit result. The ’C6202 can produce two multiply-accumulates (MACs) per cycle for a total of 500 million MACs per second (MMACS). The ’C6202 DSP also has application-specific hardware logic, on-chip memory, and additional on-chip peripherals.

The ’C6202 includes a large bank of on-chip memory and has a powerful and diverse set of peripherals. Program memory consists of two 128K-byte blocks, with one block configured as memory-mapped program space, and the other block user-configured as cache or memory-mapped program space. Data memory consists of two 64K-byte blocks of RAM. The peripheral set includes three multichannel buffered serial ports (McBSPs), two general-purpose timers, an expansion bus (XB) that offers ease of interface to synchronous or asynchronous industry-standard host bus protocols, and a glueless external memory interface (EMIF) capable of interfacing to SDRAM or SBSRAM and asynchronous peripherals.

The ’C6202 has a complete set of development tools which includes: a new C compiler, an assembly optimizer to simplify programming and scheduling, and a Windows™ debugger interface for visibility into source code execution.

## device characteristics

Table 1 provides an overview of the ’C6202 DSP. The table shows significant features of each device, including the capacity of on-chip RAM, the peripherals, the execution time, and the package type with pin count.

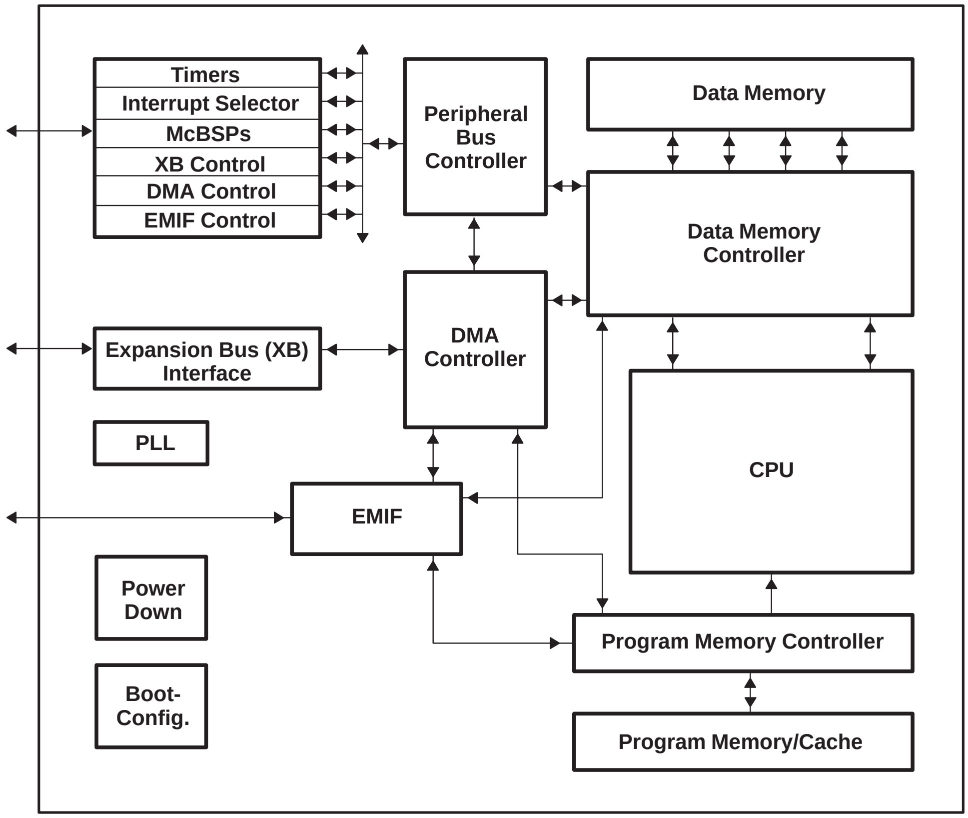
**Table 1. Characteristics of the ’C6202 Processors**

| CHARACTERISTICS | DESCRIPTION |
|---|---|
| Device Number | TMS320C6202 |
| On-Chip Memory | 2 Mbit Program Memory<br>(organized as 2 blocks)<br>1 Mbit Data Memory<br>(organized as 2 blocks) |
| Peripherals | 3 Multichannel Buffered Serial Ports (McBSP)<br>2 General-Purpose Timers<br>External Memory Interface (EMIF)<br>Expansion Bus (XB) |
| Cycle Time | 4 ns |
| Package Type | 27 mm × 27 mm, 352-Pin BGA (GJL)<br>18 mm × 18 mm, 384-Pin BGA (GLS) |
| Nominal Voltage | 1.8 V Core<br>3.3 V I/O |

ADVANCE INFORMATION

TI is a trademark of Texas Instruments Incorporated.
Windows is a registered trademark of the Microsoft Corporation.

### functional block diagram




ADVANCE INFORMATION

## CPU description

The CPU fetches VelociTI advanced very-long instruction words (VLIW) (256 bits wide) to supply up to eight 32-bit instructions to the eight functional units during every clock cycle. The VelociTI VLIW architecture features controls by which all eight units do not have to be supplied with instructions if they are not ready to execute. The first bit of every 32-bit instruction determines if the next instruction belongs to the same execute packet as the previous instruction, or whether it should be executed in the following clock as a part of the next execute packet. Fetch packets are always 256 bits wide; however, the execute packets can vary in size. The variable-length execute packets are a key memory-saving feature, distinguishing the ’C6200 CPU from other VLIW architectures.

The CPU features two sets of functional units. Each set contains four units and a register file. One set contains functional units .L1, .S1, .M1, and .D1; the other set contains units .D2, .M2, .S2, and .L2. The two register files each contain 16 32-bit registers for a total of 32 general-purpose registers. The two sets of functional units, along with two register files, compose sides A and B of the CPU (see Figure 1 and Figure 2). The four functional units on each side of the CPU can freely share the 16 registers belonging to that side. Additionally, each side features a single data bus connected to all the registers on the other side, by which the two sets of functional units can access data from the register files on the opposite side. While register access by functional units on the same side of the CPU as the register file can service all the units in a single clock cycle, register access using the register file across the CPU supports one read and one write per cycle.

Another key feature of the ’C6200 CPU is the load/store architecture, where all instructions operate on registers (as opposed to data in memory). Two sets of data-addressing units (.D1 and .D2) are responsible for all data transfers between the register files and the memory. The data address driven by the .D units allows data addresses generated from one register file to be used to load or store data to or from the other register file. The ’C6200 CPU supports a variety of indirect addressing modes using either linear- or circular-addressing modes with 5- or 15-bit offsets. All instructions are conditional, and most can access any one of the 32 registers. Some registers, however, are singled out to support specific addressing or to hold the condition for conditional instructions (if the condition is not automatically “true”). The two .M functional units are dedicated for multiplies. The two .S and .L functional units perform a general set of arithmetic, logical, and branch functions with results available every clock cycle.

The processing flow begins when a 256-bit-wide instruction fetch packet is fetched from a program memory. The 32-bit instructions destined for the individual functional units are “linked” together by “1” bits in the least significant bit (LSB) position of the instructions. The instructions that are “chained” together for simultaneous execution (up to eight in total) compose an execute packet. A “0” in the LSB of an instruction breaks the chain, effectively placing the instructions that follow it in the next execute packet. If an execute packet crosses the fetch-packet boundary (256 bits wide), the assembler places it in the next fetch packet, while the remainder of the current fetch packet is padded with NOP instructions. The number of execute packets within a fetch packet can vary from one to eight. Execute packets are dispatched to their respective functional units at the rate of one per clock cycle and the next 256-bit fetch packet is not fetched until all the execute packets from the current fetch packet have been dispatched. After decoding, the instructions simultaneously drive all active functional units for a maximum execution rate of eight instructions every clock cycle. While most results are stored in 32-bit registers, they can be subsequently moved to memory as bytes or half-words as well. All load and store instructions are byte-, half-word, or word-addressable.

## CPU description (continued)

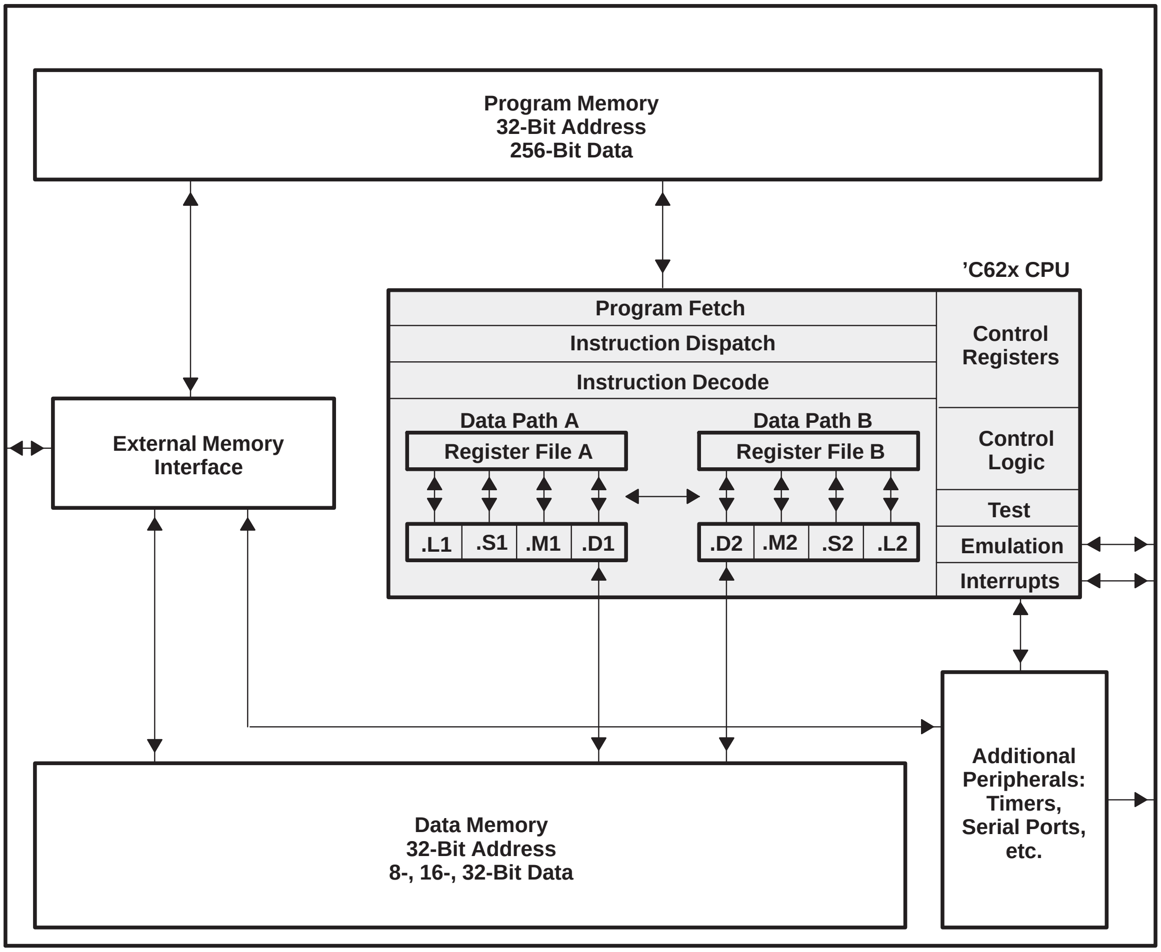



**Figure 1. TMS320C62x CPU Block Diagram**

ADVANCE INFORMATION

## CPU description (continued)

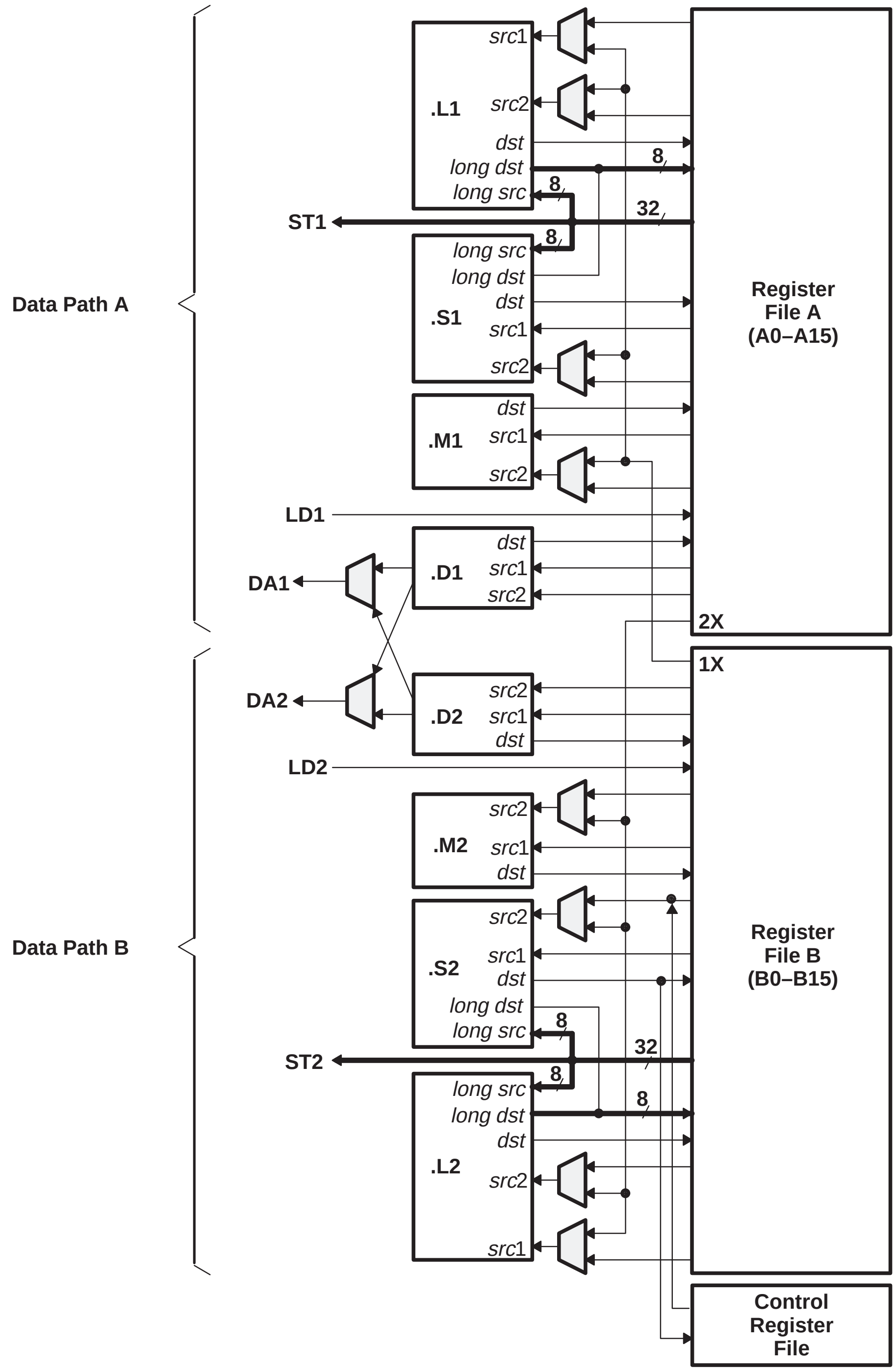
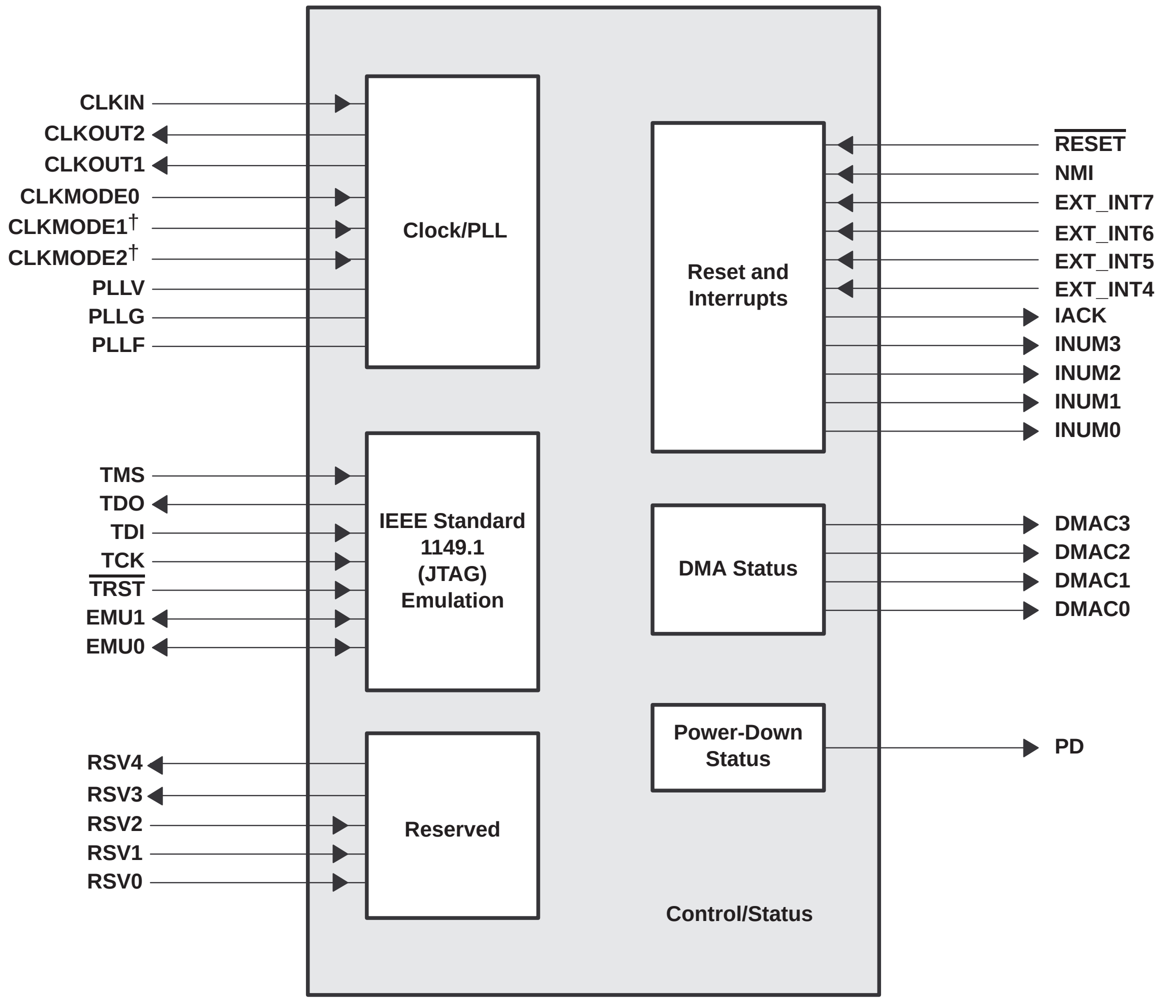



**Figure 2. TMS320C62x CPU Data Paths**

ADVANCE INFORMATION

## signal groups description




† For GLS devices only

**Figure 3. CPU Signals**

ADVANCE INFORMATION

## signal groups description (continued)

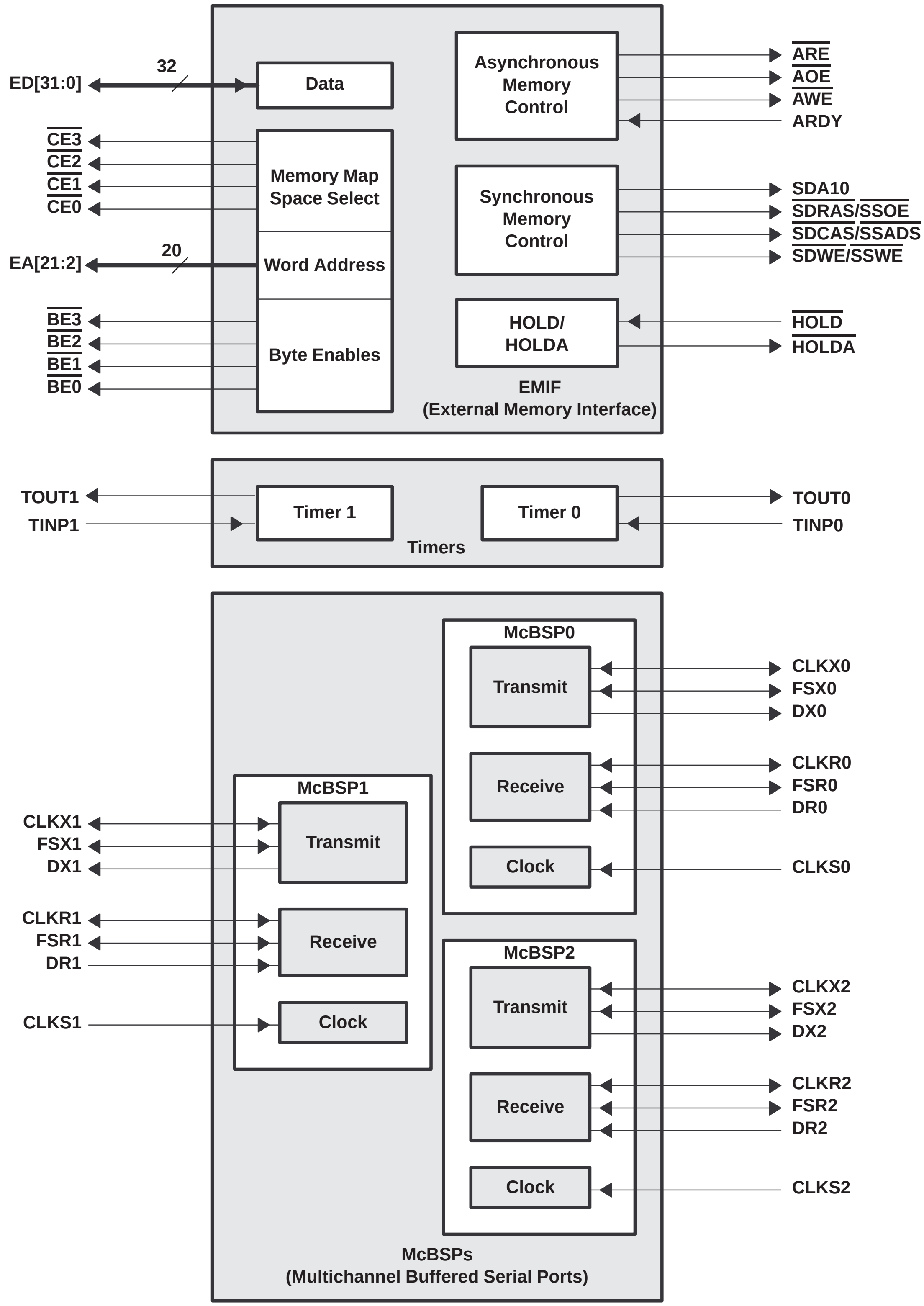



**Figure 4. Peripheral Signals**

ADVANCE INFORMATION

## signal groups description (continued)

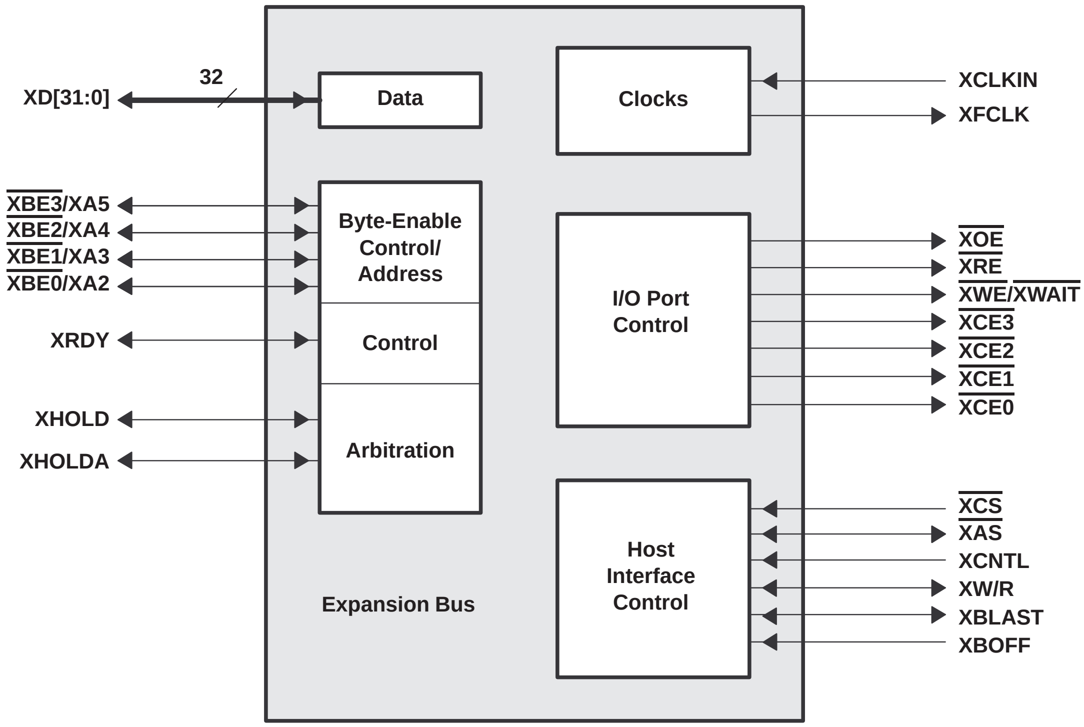



**Figure 4. Peripheral Signals (Continued)**

ADVANCE INFORMATION

## Signal Descriptions

| SIGNAL NAME | PIN NO. GJL | PIN NO. GLS | TYPE† | DESCRIPTION |
|---|---|---|---|---|
| **CLOCK/PLL** | | | | |
| CLKIN | C12 | B10 | I | Clock Input |
| CLKOUT1 | AD20 | Y18 | O | Clock output at full device speed |
| CLKOUT2 | AC19 | AB19 | O | Clock output at half of device speed<br>• Used for synchronous memory interface |
| CLKMODE0 | B15 | B12 | I | Clock mode selects (Note: CLKMODE1 and CLKMODE2 selects are for GLS devices only)<br>• Selects whether the CPU clock frequency = input clock frequency x4 or x1 |
| CLKMODE1 | – | A9 | I | |
| CLKMODE2 | – | A14 | I | |
| PLLV‡ | D13 | C11 | A§ | PLL analog $V_{CC}$ connection for the low-pass filter |
| PLLG‡ | D14 | C12 | A§ | PLL analog GND connection for the low-pass filter |
| PLLF | C13 | A11 | A§ | PLL low-pass filter connection to external components and a bypass capacitor |
| **JTAG EMULATION** | | | | |
| TMS | AD7 | Y5 | I | JTAG test-port mode select (features an internal pullup) |
| TDO | AE6 | AA4 | O/Z | JTAG test-port data out |
| TDI | AF5 | Y4 | I | JTAG test-port data in (features an internal pullup) |
| TCK | AE5 | AB2 | I | JTAG test-port clock |
| $\overline{\text{TRST}}$ | AC7 | AA3 | I | JTAG test-port reset (features an internal pulldown) |
| EMU1 | AF6 | AA5 | I/O/Z | Emulation pin 1, pullup with a dedicated 20-kΩ resistor¶ |
| EMU0 | AC8 | AB4 | I/O/Z | Emulation pin 0, pullup with a dedicated 20-kΩ resistor¶ |
| **RESET AND INTERRUPTS** | | | | |
| $\overline{\text{RESET}}$ | K2 | J3 | I | Device reset |
| NMI | L2 | K2 | I | Nonmaskable interrupt<br>• Edge-driven (rising edge) |
| EXT_INT7 | V4 | U2 | I | External interrupts<br>• Edge-driven (rising edge) |
| EXT_INT6 | Y2 | U3 | | |
| EXT_INT5 | AA1 | W1 | | |
| EXT_INT4 | W4 | V2 | | |
| IACK | Y1 | V1 | O | Interrupt acknowledge for all active interrupts serviced by the CPU |
| INUM3 | V2 | R3 | O | Active interrupt identification number<br>• Valid during IACK for all active interrupts (not just external)<br>• Encoding order follows the interrupt-service fetch-packet ordering |
| INUM2 | U4 | T1 | | |
| INUM1 | V3 | T2 | | |
| INUM0 | W2 | T3 | | |
| **POWER-DOWN STATUS** | | | | |
| PD | AB2 | Y2 | O | Power-down modes 2 or 3 (active if high) |

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground
‡ PLLV and PLLG are not part of external voltage supply or ground. See the *clock PLL* section for information on how to connect these pins.
§ A = Analog Signal (PLL Filter)
¶ For emulation and normal operation, pull up EMU1 and EMU0 with a dedicated 20-kΩ resistor. For boundary scan, pull down EMU1 and EMU0 with a dedicated 20-kΩ resistor.

ADVANCE INFORMATION

## Signal Descriptions (Continued)

| SIGNAL NAME | PIN NO. | | TYPE† | DESCRIPTION |
|---|---|---|---|---|
| | GJL | GLS | | |
| **EXPANSION BUS** | | | | |
| XCLKIN | A9 | C8 | I | Expansion bus synchronous host interface clock input |
| XFCLK | B9 | A8 | O | Expansion bus FIFO interface clock output |
| XD31 | D15 | C13 | I/O/Z | Expansion bus data<br>• Used for transfer of data, address, and control<br>• Also controls initialization of DSP modes and expansion bus at reset via pullup/pulldown resistors<br>– XCE[3:0] memory type<br>– XBLAST polarity<br>– XW/R polarity<br>– Asynchronous or synchronous host operation<br>– Arbitration mode (internal or external)<br>– FIFO mode<br>– Little endian/big endian<br>– Boot mode |
| XD30 | B16 | A13 | | |
| XD29 | A17 | C14 | | |
| XD28 | B17 | B14 | | |
| XD27 | D16 | B15 | | |
| XD26 | A18 | C15 | | |
| XD25 | B18 | A15 | | |
| XD24 | D17 | B16 | | |
| XD23 | C18 | C16 | | |
| XD22 | A20 | A17 | | |
| XD21 | D18 | B17 | | |
| XD20 | C19 | C17 | | |
| XD19 | A21 | B18 | | |
| XD18 | D19 | A19 | | |
| XD17 | C20 | C18 | | |
| XD16 | B21 | B19 | | |
| XD15 | A22 | C19 | | |
| XD14 | D20 | B20 | | |
| XD13 | B22 | A21 | | |
| XD12 | E25 | C21 | | |
| XD11 | F24 | D20 | | |
| XD10 | E26 | B22 | | |
| XD9 | F25 | D21 | | |
| XD8 | G24 | E20 | | |
| XD7 | H23 | E21 | | |
| XD6 | F26 | D22 | | |
| XD5 | G25 | F20 | | |
| XD4 | J23 | F21 | | |
| XD3 | G26 | E22 | | |
| XD2 | H25 | G20 | | |
| XD1 | J24 | G21 | | |
| XD0 | K23 | G22 | | |
| $\overline{\text{XCE3}}$ | F2 | D2 | O/Z | Expansion bus I/O port memory space enables<br>• Enabled by bits 28, 29, and 30 of the word address<br>• Only one asserted during any I/O port data access |
| $\overline{\text{XCE2}}$ | E1 | B1 | | |
| $\overline{\text{XCE1}}$ | F3 | D3 | | |
| $\overline{\text{XCE0}}$ | E2 | C2 | | |

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

## Signal Descriptions (Continued)

| SIGNAL NAME | PIN NO. | | TYPE† | DESCRIPTION |
|---|---|---|---|---|
| | GJL | GLS | | |
| **EXPANSION BUS (CONTINUED)** | | | | |
| XBE3/XA5 | C7 | C5 | I/O/Z | Expansion bus multiplexed byte-enable control/address signals<br>• Act as byte enable for host port operation<br>• Act as address for I/O port operation |
| XBE2/XA4 | D8 | A4 | | |
| XBE1/XA3 | A6 | B5 | | |
| XBE0/XA2 | C8 | C6 | | |
| XOE | A7 | A6 | O/Z | Expansion bus I/O port output enable |
| XRE | C9 | C7 | O/Z | Expansion bus I/O port read enable |
| XWE/XWAIT | D10 | B7 | O/Z | Expansion bus I/O port write enable and host port wait signals |
| XCS | A10 | C9 | I | Expansion bus host port chip-select input |
| XAS | D9 | B6 | I/O/Z | Expansion bus host port address strobe |
| XCNTL | B10 | B9 | I | Expansion bus host control. XCNTL selects between expansion bus address or data register |
| XW/R | D11 | B8 | I/O/Z | Expansion bus host port write/read enable. XW/R polarity selected at reset |
| XRDY | A5 | C4 | I/O/Z | Expansion bus host port ready (active low) and I/O port ready (active high) |
| XBLAST | B6 | B4 | I/O/Z | Expansion bus host port burst last–polarity selected at reset |
| XBOFF | B11 | A10 | I | Expansion bus back off |
| XHOLD | B5 | A2 | I/O/Z | Expansion bus hold request |
| XHOLDA | D7 | B3 | I/O/Z | Expansion bus hold acknowledge |
| **EMIF – CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY** | | | | |
| CE3 | AB25 | Y21 | O/Z | Memory space enables<br>• Enabled by bits 24 and 25 of the word address<br>• Only one asserted during any external data access |
| CE2 | AA24 | W20 | | |
| CE1 | AB26 | AA22 | | |
| CE0 | AA25 | W21 | | |
| BE3 | Y24 | V20 | O/Z | Byte-enable control<br>• Decoded from the two lowest bits of the internal address<br>• Byte-write enables for most types of memory<br>• Can be directly connected to SDRAM read and write mask signal (SDQM) |
| BE2 | W23 | V21 | | |
| BE1 | AA26 | W22 | | |
| BE0 | Y25 | U20 | | |
| **EMIF – ADDRESS** | | | | |
| EA21 | J25 | H20 | O/Z | External address (word address) |
| EA20 | J26 | H21 | | |
| EA19 | L23 | H22 | | |
| EA18 | K25 | J20 | | |
| EA17 | L24 | J21 | | |
| EA16 | L25 | K21 | | |
| EA15 | M23 | K20 | | |
| EA14 | M24 | K22 | | |
| EA13 | M25 | L21 | | |
| EA12 | N23 | L20 | | |
| EA11 | P24 | L22 | | |

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

ADVANCE INFORMATION

ADVANCE INFORMATION

## Signal Descriptions (Continued)

| SIGNAL NAME | PIN NO. GJL | PIN NO. GLS | TYPE† | DESCRIPTION |
|---|---|---|---|---|
| **EMIF – ADDRESS (CONTINUED)** | | | | |
| EA10 | P23 | M20 | O/Z | External address (word address) |
| EA9 | R25 | M21 | | |
| EA8 | R24 | N22 | | |
| EA7 | R23 | N20 | | |
| EA6 | T25 | N21 | | |
| EA5 | T24 | P21 | | |
| EA4 | U25 | P20 | | |
| EA3 | T23 | R22 | | |
| EA2 | V26 | R21 | | |
| **EMIF – DATA** | | | | |
| ED31 | AD8 | Y6 | I/O/Z | External data |
| ED30 | AC9 | AA6 | | |
| ED29 | AF7 | AB6 | | |
| ED28 | AD9 | Y7 | | |
| ED27 | AC10 | AA7 | | |
| ED26 | AE9 | AB8 | | |
| ED25 | AF9 | Y8 | | |
| ED24 | AC11 | AA8 | | |
| ED23 | AE10 | AA9 | | |
| ED22 | AD11 | Y9 | | |
| ED21 | AE11 | AB10 | | |
| ED20 | AC12 | Y10 | | |
| ED19 | AD12 | AA10 | | |
| ED18 | AE12 | AA11 | | |
| ED17 | AC13 | Y11 | | |
| ED16 | AD14 | AB12 | | |
| ED15 | AC14 | Y12 | | |
| ED14 | AE15 | AA12 | | |
| ED13 | AD15 | AA13 | | |
| ED12 | AC15 | Y13 | | |
| ED11 | AE16 | AB13 | | |
| ED10 | AD16 | Y14 | | |
| ED9 | AE17 | AA14 | | |
| ED8 | AC16 | AA15 | | |
| ED7 | AF18 | Y15 | | |
| ED6 | AE18 | AB15 | | |
| ED5 | AC17 | AA16 | | |
| ED4 | AD18 | Y16 | | |
| ED3 | AF20 | AB17 | | |
| ED2 | AC18 | AA17 | | |
| ED1 | AD19 | Y17 | | |
| ED0 | AF21 | AA18 | | |

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

## Signal Descriptions (Continued)

| SIGNAL NAME | PIN NO. | | TYPE† | DESCRIPTION |
|---|---|---|---|---|
| | GJL | GLS | | |
| **EMIF – ASYNCHRONOUS MEMORY CONTROL** | | | | |
| $\overline{ARE}$ | V24 | T21 | O/Z | Asynchronous memory read enable |
| $\overline{AOE}$ | V25 | R20 | O/Z | Asynchronous memory output enable |
| $\overline{AWE}$ | U23 | T22 | O/Z | Asynchronous memory write enable |
| ARDY | W25 | T20 | I | Asynchronous memory ready input |
| **EMIF – SYNCHRONOUS DRAM (SDRAM)/SYNCHRONOUS BURST SRAM (SBSRAM) CONTROL** | | | | |
| SDA10 | AE21 | AA19 | O/Z | SDRAM address 10 (separate for deactivate command) |
| $\overline{SDCAS}/\overline{SSADS}$ | AE22 | AB21 | O/Z | SDRAM column-address strobe/SBSRAM address strobe |
| $\overline{SDRAS}/\overline{SSOE}$ | AF22 | Y19 | O/Z | SDRAM row-address strobe/SBSRAM output enable |
| $\overline{SDWE}/\overline{SSWE}$ | AC20 | AA20 | O/Z | SDRAM write enable/SBSRAM write enable |
| **EMIF – BUS ARBITRATION** | | | | |
| $\overline{HOLD}$ | Y26 | V22 | I | Hold request from the host |
| $\overline{HOLDA}$ | V23 | U21 | O | Hold-request-acknowledge to the host |
| **TIMERS** | | | | |
| TOUT1 | J4 | F2 | O | Timer 1 or general-purpose output |
| TINP1 | G2 | F3 | I | Timer 1 or general-purpose input |
| TOUT0 | F1 | D1 | O | Timer 0 or general-purpose output |
| TINP0 | H4 | E2 | I | Timer 0 or general-purpose input |
| **DMA ACTION COMPLETE STATUS** | | | | |
| DMAC3 | Y3 | V3 | O | DMA action complete |
| DMAC2 | AA2 | W2 | | |
| DMAC1 | AB1 | AA1 | | |
| DMAC0 | AA3 | W3 | | |
| **MULTICHANNEL BUFFERED SERIAL PORT 0 (McBSP0)** | | | | |
| CLKS0 | M4 | K3 | I | External clock source (as opposed to internal) |
| CLKR0 | M2 | L2 | I/O/Z | Receive clock |
| CLKX0 | M3 | K1 | I/O/Z | Transmit clock |
| DR0 | R2 | M2 | I | Receive data |
| DX0 | P4 | M3 | O/Z | Transmit data |
| FSR0 | N3 | M1 | I/O/Z | Receive frame sync |
| FSX0 | N4 | L3 | I/O/Z | Transmit frame sync |
| **MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP1)** | | | | |
| CLKS1 | G1 | E1 | I | External clock source (as opposed to internal) |
| CLKR1 | J3 | G2 | I/O/Z | Receive clock |
| CLKX1 | H2 | G3 | I/O/Z | Transmit clock |
| DR1 | L4 | H1 | I | Receive data |
| DX1 | J1 | H2 | O/Z | Transmit data |
| FSR1 | J2 | H3 | I/O/Z | Receive frame sync |
| FSX1 | K4 | G1 | I/O/Z | Transmit frame sync |

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

ADVANCE INFORMATION

ADVANCE INFORMATION

### Signal Descriptions (Continued)

| SIGNAL NAME | PIN NO. GJL | PIN NO. GLS | TYPE† | DESCRIPTION |
|---|---|---|---|---|
| **MULTICHANNEL BUFFERED SERIAL PORT 2 (McBSP2)** | | | | |
| CLKS2 | R3 | N1 | I | External clock source (as opposed to internal) |
| CLKR2 | T2 | N2 | I/O/Z | Receive clock |
| CLKX2 | R4 | N3 | I/O/Z | Transmit clock |
| DR2 | V1 | R2 | I | Receive data |
| DX2 | T4 | R1 | O/Z | Transmit data |
| FSR2 | U2 | P3 | I/O/Z | Receive frame sync |
| FSX2 | T3 | P2 | I/O/Z | Transmit frame sync |
| **RESERVED FOR TEST** | | | | |
| RSV0 | L3 | J2 | I | Reserved for testing, pullup with a dedicated 20-kΩ resistor |
| RSV1 | G3 | E3 | I | Reserved for testing, pullup with a dedicated 20-kΩ resistor |
| RSV2 | A12 | B11 | I | Reserved for testing, pullup with a dedicated 20-kΩ resistor |
| RSV3 | C15 | B13 | O | Reserved (leave unconnected, ***do not*** connect to power or ground) |
| RSV4 | D12 | C10 | O | Reserved (leave unconnected, ***do not*** connect to power or ground) |
| **SUPPLY VOLTAGE PINS** | | | | |
| $DV_{DD}$ | A11 | A3 | S | 3.3-V supply voltage |
| | A16 | A7 | | |
| | B7 | A16 | | |
| | B8 | A20 | | |
| | B19 | D4 | | |
| | B20 | D6 | | |
| | C6 | D7 | | |
| | C10 | D9 | | |
| | C14 | D10 | | |
| | C17 | D13 | | |
| | C21 | D14 | | |
| | G4 | D16 | | |
| | G23 | D17 | | |
| | H3 | D19 | | |
| | H24 | F1 | | |
| | K3 | F4 | | |
| | K24 | F19 | | |
| | L1 | F22 | | |
| | L26 | G4 | | |
| | N24 | G19 | | |
| | P3 | J4 | | |

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

### Signal Descriptions (Continued)

| SIGNAL NAME | PIN NO. | | TYPE† | DESCRIPTION |
|---|---|---|---|---|
| | GJL | GLS | | |
| **SUPPLY VOLTAGE PINS (CONTINUED)** | | | | |
| $DV_{DD}$ | T1 | J19 | S | 3.3-V supply voltage |
| | T26 | K4 | | |
| | U3 | K19 | | |
| | U24 | L1 | | |
| | W3 | M22 | | |
| | W24 | N4 | | |
| | Y4 | N19 | | |
| | Y23 | P4 | | |
| | AD6 | P19 | | |
| | AD10 | T4 | | |
| | AD13 | T19 | | |
| | AD17 | U1 | | |
| | AD21 | U4 | | |
| | AE7 | U19 | | |
| | AE8 | U22 | | |
| | AE19 | W4 | | |
| | AE20 | W6 | | |
| | AF11 | W7 | | |
| | AF16 | W9 | | |
| | – | W10 | | |
| | – | W13 | | |
| | – | W14 | | |
| | – | W16 | | |
| | – | W17 | | |
| | – | W19 | | |
| | – | AB5 | | |
| | – | AB9 | | |
| | – | AB14 | | |
| | – | AB18 | | |
| $CV_{DD}$ | A1 | E7 | S | 1.8-V supply voltage |
| | A2 | E8 | | |
| | A3 | E10 | | |
| | A24 | E11 | | |
| | A25 | E12 | | |
| | A26 | E13 | | |
| | B1 | E15 | | |
| | B2 | E16 | | |
| | B3 | F7 | | |
| | B24 | F8 | | |
| | B25 | F9 | | |
| | B26 | F11 | | |
| | C1 | F12 | | |

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

ADVANCE INFORMATION

ADVANCE INFORMATION

## Signal Descriptions (Continued)

| SIGNAL NAME | PIN NO. GJL | PIN NO. GLS | TYPE† | DESCRIPTION |
|---|---|---|---|---|
| SUPPLY VOLTAGE PINS (CONTINUED) | | | | |
| $CV_{DD}$ | C2 | F14 | S | 1.8-V supply voltage |
| | C3 | F15 | | |
| | C4 | F16 | | |
| | C23 | G5 | | |
| | C24 | G6 | | |
| | C25 | G17 | | |
| | C26 | G18 | | |
| | D3 | H5 | | |
| | D4 | H6 | | |
| | D5 | H17 | | |
| | D22 | H18 | | |
| | D23 | J6 | | |
| | D24 | J17 | | |
| | E4 | K5 | | |
| | E23 | K18 | | |
| | AB4 | L5 | | |
| | AB23 | L6 | | |
| | AC3 | L17 | | |
| | AC4 | L18 | | |
| | AC5 | M5 | | |
| | AC22 | M6 | | |
| | AC23 | M17 | | |
| | AC24 | M18 | | |
| | AD1 | N5 | | |
| | AD2 | N18 | | |
| | AD3 | P6 | | |
| | AD4 | P17 | | |
| | AD23 | R5 | | |
| | AD24 | R6 | | |
| | AD25 | R17 | | |
| | AD26 | R18 | | |
| | AE1 | T5 | | |
| | AE2 | T6 | | |
| | AE3 | T17 | | |
| | AE24 | T18 | | |
| | AE25 | U7 | | |
| | AE26 | U8 | | |
| | AF1 | U9 | | |
| | AF2 | U11 | | |
| | AF3 | U12 | | |
| | AF24 | U14 | | |
| | AF25 | U15 | | |

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

### Signal Descriptions (Continued)

| SIGNAL NAME | PIN NO. GJL | PIN NO. GLS | TYPE† | DESCRIPTION |
|---|---|---|---|---|
| **SUPPLY VOLTAGE PINS (CONTINUED)** | | | | |
| $CV_{DD}$ | AF26 | U16 | S | 1.8-V supply voltage |
| | – | V7 | | |
| | – | V8 | | |
| | – | V10 | | |
| | – | V11 | | |
| | – | V12 | | |
| | – | V13 | | |
| | – | V15 | | |
| | – | V16 | | |
| **GROUND PINS** | | | | |
| $V_{SS}$ | A4 | A1 | GND | Ground pins |
| | A8 | A5 | | |
| | A13 | A12 | | |
| | A14 | A18 | | |
| | A15 | A22 | | |
| | A19 | B2 | | |
| | A23 | B21 | | |
| | B4 | C1 | | |
| | B12 | C3 | | |
| | B13 | C20 | | |
| | B14 | C22 | | |
| | B23 | D5 | | |
| | C5 | D8 | | |
| | C11 | D11 | | |
| | C16 | D12 | | |
| | C22 | D15 | | |
| | D1 | D18 | | |
| | D2 | E4 | | |
| | D6 | E5 | | |
| | D21 | E6 | | |
| | D25 | E9 | | |
| | D26 | E14 | | |
| | E3 | E17 | | |
| | E24 | E18 | | |
| | F4 | E19 | | |
| | F23 | F5 | | |
| | H1 | F6 | | |
| | H26 | F10 | | |

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

ADVANCE INFORMATION

ADVANCE INFORMATION

## Signal Descriptions (Continued)

| SIGNAL NAME | PIN NO. | | TYPE† | DESCRIPTION |
|---|---|---|---|---|
| | GJL | GLS | | |
| **GROUND PINS (CONTINUED)** | | | | |
| $V_{SS}$ | K1 | F13 | GND | Ground pins |
| | K26 | F17 | | |
| | M1 | F18 | | |
| | M26 | H4 | | |
| | N1 | H19 | | |
| | N2 | J1 | | |
| | N25 | J5 | | |
| | N26 | J18 | | |
| | P1 | J22 | | |
| | P2 | K6 | | |
| | P25 | K17 | | |
| | P26 | L4 | | |
| | R1 | L19 | | |
| | R26 | M4 | | |
| | U1 | M19 | | |
| | U26 | N6 | | |
| | W1 | N17 | | |
| | W26 | P1 | | |
| | AA4 | P5 | | |
| | AA23 | P18 | | |
| | AB3 | P22 | | |
| | AB24 | R4 | | |
| | AC1 | R19 | | |
| | AC2 | U5 | | |
| | AC6 | U6 | | |
| | AC21 | U10 | | |
| | AC25 | U13 | | |
| | AC26 | U17 | | |
| | AD5 | U18 | | |
| | AD22 | V4 | | |
| | AE4 | V5 | | |
| | AE13 | V6 | | |
| | AE14 | V9 | | |
| | AE23 | V14 | | |
| | AF4 | V17 | | |
| | AF8 | V18 | | |

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

ADVANCE INFORMATION

### Signal Descriptions (Continued)

| SIGNAL NAME | PIN NO. GJL | PIN NO. GLS | TYPE† | DESCRIPTION |
|---|---|---|---|---|
| GROUND PINS (CONTINUED) | | | | |
| $V_{SS}$ | AF10 | V19 | GND | Ground pins |
| | AF12 | W5 | | |
| | AF13 | W8 | | |
| | AF14 | W11 | | |
| | AF15 | W12 | | |
| | AF17 | W15 | | |
| | AF19 | W18 | | |
| | AF23 | Y1 | | |
| | – | Y3 | | |
| | – | Y20 | | |
| | – | Y22 | | |
| | – | AA2 | | |
| | – | AA21 | | |
| | – | AB1 | | |
| | – | AB3 | | |
| | – | AB7 | | |
| | – | AB11 | | |
| | – | AB16 | | |
| | – | AB20 | | |
| | – | AB22 | | |

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

## development support

Texas Instruments offers an extensive line of development tools for the ’C6200 generation of DSPs, including tools to evaluate the performance of the processors, generate code, develop algorithm implementations, and fully integrate and debug software and hardware modules.

The following products support development of ’C6200-based applications:

**Software Development Tools:**
Assembly optimizer
Assembler/Linker
Simulator
Optimizing ANSI C compiler
Application algorithms
C/Assembly debugger and code profiler

**Hardware Development Tools:**
Extended development system (XDS™) emulator (supports ’C6200 multiprocessor system debug)
EVM (Evaluation Module)

The *TMS320 DSP Development Support Reference Guide* (SPRU011) contains information about development-support products for all TMS320 family member devices, including documentation. See this document for further information on TMS320 documentation or any TMS320 support products from Texas Instruments. An additional document, the *TMS320 Third-Party Support Reference Guide* (SPRU052), contains information about TMS320-related products from other companies in the industry. To receive TMS320 literature, contact the Literature Response Center at 800/477-8924.

See Table 2 for a complete listing of development-support tools for the ’C6200. For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

**Table 2. TMS320C6xx Development-Support Tools**

| DEVELOPMENT TOOL | PLATFORM | PART NUMBER |
|---|---|---|
| **Software** | | |
| C Compiler/Assembler/Linker/Assembly Optimizer | Win32™ | TMDX3246855-07 |
| C Compiler/Assembler/Linker/Assembly Optimizer | SPARC™ Solaris™ | TMDX3246555-07 |
| Simulator | Win32 | TMDS3246851-07 |
| Simulator | SPARC Solaris | TMDS3246551-07 |
| XDS510™ Debugger/Emulation Software | Win32, Windows NT™ | TMDX324016X-07 |
| **Hardware** | | |
| XDS510 Emulator† | PC | TMDS00510 |
| XDS510WS™ Emulator‡ | SCSI | TMDS00510WS |
| **Software/Hardware** | | |
| EVM Evaluation Kit | PC/Win95/Windows NT | TMDX3260A6201 |
| EVM Evaluation Kit (including TMDX3246855–07) | PC/Win95/Windows NT | TMDX326006201 |

† Includes XDS510 board and JTAG emulation cable. TMDX324016X-07 C-source Debugger/Emulation software is not included.
‡ Includes XDS510WS box, SCSI cable, power supply, and JTAG emulation cable.

XDS, XDS510, and XDS510WS are trademarks of Texas Instruments Incorporated.
Win32 and Windows NT are trademarks of Microsoft Corporation.
SPARC is a trademark of SPARC International, Inc.
Solaris is a trademark of Sun Microsystems, Inc.

### device and development-support tool nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all TMS320 devices and support tools. Each TMS320 member has one of three prefixes: TMX, TMP, or TMS. Texas Instruments recommends two of three possible prefix designators for support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX / TMDX) through fully qualified production devices/tools (TMS / TMDS).

Device development evolutionary flow:

**TMX** Experimental device that is not necessarily representative of the final device's electrical specifications

**TMP** Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification

**TMS** Fully qualified production device

Support tool development evolutionary flow:

**TMDX** Development-support product that has not yet completed Texas Instruments internal qualification testing.

**TMDS** Fully qualified development-support product

TMX and TMP devices and TMDX development-support tools are shipped against the following disclaimer:

"Developmental product is intended for internal evaluation purposes."

TMS devices and TMDS development-support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

Predictions show that prototype devices ( TMX or TMP) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, GJL), the temperature range (for example, blank is the default commercial temperature range), and the device speed range in megahertz (for example, -250 is 250 MHz). Figure 5 provides a legend for reading the complete device name for any TMS320 family member.

## device and development-support tool nomenclature (continued)

**TMS 320 C 6202 GJL (A) –250**

**PREFIX**
- TMX = Experimental device
- TMP = Prototype device
- TMS = Qualified device
- SMJ = MIL-STD-883C
- SM = High Rel (non-883C)

**DEVICE FAMILY**
- 320 = TMS320 family

**TECHNOLOGY**
- C = CMOS
- E = CMOS EPROM
- F = CMOS Flash EEPROM

**DEVICE SPEED RANGE**
- –100 MHz
- –150 MHz
- –167 MHz
- –200 MHz
- –233 MHz
- –250 MHz
- –300 MHz

**TEMPERATURE RANGE (DEFAULT: 0°C TO 90°C)**
- Blank = 0°C to 90°C, commercial temperature
- A = –40°C to 105°C, extended temperature

**PACKAGE TYPE†**
- N = Plastic DIP
- J = Ceramic DIP
- JD = Ceramic DIP side-brazed
- GB = Ceramic PGA
- FZ = Ceramic CC
- FN = Plastic leaded CC
- FD = Ceramic leadless CC
- PJ = 100-pin plastic EIAJ QFP
- PQ = 132-pin plastic bumpered QFP
- PZ = 100-pin plastic TQFP
- PBK = 128-pin plastic TQFP
- PGE = 144-pin plastic TQFP
- GFN = 256-pin plastic BGA
- GGU = 144-pin plastic BGA
- GGP = 352-pin plastic BGA
- GJC = 352-pin plastic BGA
- GJL = 352-pin plastic BGA
- GLS = 384-pin plastic BGA

**DEVICE**
- ’1x DSP: 10, 16, 14, 17, 15
- ’2x DSP: 25, 26
- ’2xx DSP: 203, 206, 240, 204, 209
- ’3x DSP: 30, 31, 32
- ’4x DSP: 40, 44
- ’5x DSP: 50, 53, 51, 56, 52, 57
- ’54x DSP: 541, 545, 542, 546, 543, 548
- ’6x DSP: 6201, 6201B, 6202, 6203, 6211, 6701, 6711

† DIP = Dual-In-Line Package
PGA = Pin Grid Array
CC = Chip Carrier
QFP = Quad Flat Package
TQFP = Thin Quad Flat Package
BGA = Ball Grid Array

**Figure 5. TMS320 Device Nomenclature (Including TMS320C6202)**

### documentation support

Extensive documentation supports all TMS320 family generations of devices from product announcement through applications development. The types of documentation available include: data sheets, such as this document, with design specifications; complete user's reference guides for all devices; technical briefs; development-support tools; and hardware and software applications. The following is a brief, descriptive list of support documentation specific to the ’C6x devices:

The *TMS320C6000 CPU and Instruction Set Reference Guide* (literature number SPRU189) describes the ’C6000 CPU architecture, instruction set, pipeline, and associated interrupts.

The *TMS320C6000 Peripherals Reference Guide* (literature number SPRU190) describes the functionality of the peripherals available on ’C6x devices, such as the external memory interface (EMIF), host-port interface (HPI), multichannel buffered serial ports (McBSPs), direct-memory-access (DMA), enhanced direct-memory-access (EDMA) controller, expansion bus (XB), clocking and phase-locked loop (PLL); and power-down modes. This guide also includes information on internal data and program memories.

The *TMS320C6000 Programmer's Guide* (literature number SPRU198) describes ways to optimize C and assembly code for ’C6x devices and includes application program examples.

The *TMS320C6x C Source Debugger User's Guide* (literature number SPRU188) describes how to invoke the ’C6x simulator and emulator versions of the C source debugger interface and discusses various aspects of the debugger, including: command entry, code execution, data management, breakpoints, profiling, and analysis.

The *TMS320C6x Peripheral Support Library Programmer's Reference* (literature number SPRU273) describes the contents of the ’C6x peripheral support library of functions and macros. It lists functions and macros both by header file and alphabetically, provides a complete description of each, and gives code examples to show how they are used.

*TMS320C6000 Assembly Language Tools User's Guide* (literature number SPRU186) describes the assembly language tools (assembler, linker, and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for the ’C6000 generation of devices.

The *TMS320C6x Evaluation Module Reference Guide* (literature number SPRU269) provides instructions for installing and operating the ’C6x evaluation module. It also includes support software documentation, application programming interfaces, and technical reference material.

*TMS320C62x Multichannel Evaluation Module User's Guide* (literature number SPRU285) provides instructions for installing and operating the ’C62x multichannel evaluation module. It also includes support software documentation, application programming interfaces, and technical reference material.

*TMS320C62x Multichannel Evaluation Module Technical Reference* (SPRU308) provides provides technical reference information for the ’C62x multichannel evaluation module (McEVM). It includes support software documentation, application programming interface references, and hardware descriptions for the ’C62x McEVM.

*TMS320C6000 DSP/BIOS User's Guide* (literature number SPRU303) describes how to use DSP/BIOS tools and APIs to analyze embedded real-time DSP applications.

*Code Composer User's Guide* (literature number SPRU296) explains how to use the Code Composer development environment to build and debug embedded real-time DSP applications.

*Code Composer Studio Tutorial* (literature number SPRU301) introduces the Code Composer Studio integrated development environment and software tools.

ADVANCE INFORMATION

### documentation support (continued)

The *TMS320C6000 Technical Brief* (literature number SPRU197) gives an introduction to the ’C62x/C67x devices, associated development tools, and third-party support.

A series of DSP textbooks is published by Prentice-Hall and John Wiley & Sons to support DSP research and education. The TMS320 newsletter, *Details on Signal Processing*, is published quarterly and distributed to update TMS320 customers on product information. The TMS320 DSP bulletin board service (BBS) provides access to information pertaining to the TMS320 family, including documentation, source code, and object code for many DSP algorithms and utilities. The BBS can be reached at 281/274-2323.

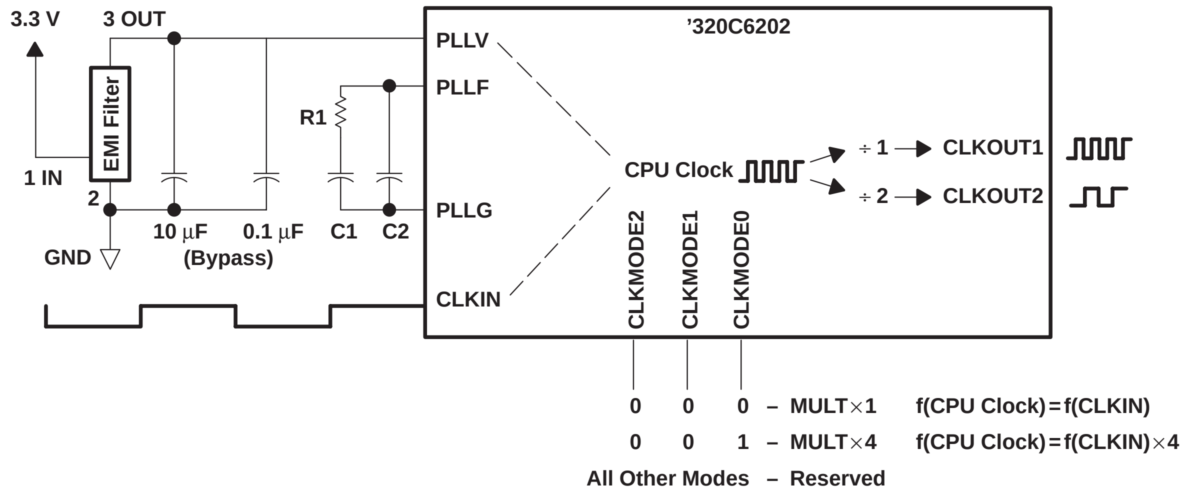
Information regarding TI DSP products is also available on the Worldwide Web at http://www.ti.com uniform resource locator (URL).

## clock PLL

All of the internal ’C6202 clocks are generated from a single source through the CLKIN pin. This source clock either drives the PLL, which generates the internal CPU clock, or bypasses the PLL to become the CPU clock.

To use the PLL to generate the CPU clock, the filter circuit shown in Figure 6 must be properly designed.

To configure the ’C6202 PLL clock for proper operation, see Figure 6 and Table 3. To minimize the clock jitter, a single clean power supply should power both the ’C6202 device and the external clock oscillator circuit. The minimum CLKIN rise and fall times should also be observed. See the *input and output clocks* section for input clock timing requirements.




NOTES: A. The ’C6202 PLL can generate CPU clock frequencies in the range of 130 MHz to 250 MHz. For frequencies below 130 MHz, the PLL should be configured to operate in bypass mode.

B. For the ’C6202, values for C1, C2, and R1 are fixed and apply to all valid frequency ranges of CLKIN and CPU clock frequency.

C. For CLKMODE x1, the PLL is bypassed and all six external PLL components can be removed. For this case, the PLLV terminal has to be connected to a clean 3.3-V supply and the PLLG and PLLF terminals should be tied together.

D. The 3.3-V supply for the EMI filter (and PLLV) must be from the same 3.3-V power plane supplying the I/O voltage, $DV_{DD}$.

E. EMI filter manufacturer TDK part number ACF451832-153-T

F. CLKMODE2 and CLKMODE1 exist only on the GLS device. There are no equivalent connections on the GJL device.

G. The reserved PLL clock modes (GLS devices only) may or may not be supported on future devices as additional PLL multiply factors. For future flexibility, a board can be designed so that these inputs are configurable (either through jumpers, switches, or 0-Ω resistors).

**Figure 6. PLL Block Diagram**

**Table 3. TMS320C6202 PLL Component Selection Table†**

| CLKMODE | CLKIN RANGE (MHz) | CPU CLOCK FREQUENCY (CLKOUT1) RANGE (MHz) | CLKOUT2 RANGE (MHz) | R1 (Ω) | C1 (nF) | C2 (pF) | TYPICAL LOCK TIME (µs) |
|---|---|---|---|---|---|---|---|
| x4 | 32.5–62.5 | 130–250 | 65–125 | 60.4 | 27 | 560 | 75 |

† Under some operating conditions, the maximum PLL lock time may vary as much as 150% from the specified typical value. For example, if the typical lock time is specified as 100 µs, the maximum value may be as long as 250 µs.

## power-supply sequencing

The 1.8-V supply powers the core and the 3.3-V supply powers the I/O buffers. The core supply should be powered up first, or at the same time as the I/O buffers supply. This is to ensure that the I/O buffers have valid inputs from the core before the output buffers are powered up, thus preventing bus contention with other chips on the board.

ADVANCE INFORMATION

ADVANCE INFORMATION

### absolute maximum ratings over operating case temperature range (unless otherwise noted)†

Supply voltage range, $CV_{DD}$ (see Note 1) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . – 0.3 V to 2.3 V
Supply voltage range, $DV_{DD}$ (see Note 1) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . –0.3 V to 4 V
Input voltage range . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . –0.3 V to 4 V
Output voltage range . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . –0.3 V to 4 V
Operating case temperature range, $T_C$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 0°C to 90°C
Storage temperature range, $T_{stg}$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . –55°C to 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to $V_{SS}$.

### recommended operating conditions

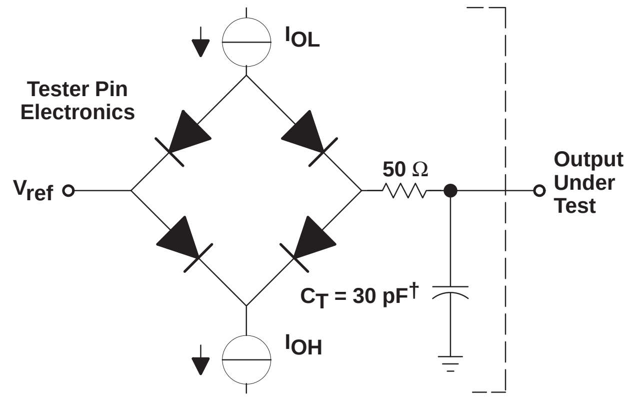
| | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|
| $CV_{DD}$ | Supply voltage | 1.71 | 1.8 | 1.89 | V |
| $DV_{DD}$ | Supply voltage | 3.14 | 3.30 | 3.46 | V |
| $V_{SS}$ | Supply ground | 0 | 0 | 0 | V |
| $V_{IH}$ | High-level input voltage | 2.0 | | | V |
| $V_{IL}$ | Low-level input voltage | | | 0.8 | V |
| $I_{OH}$ | High-level output current | | | –8 | mA |
| $I_{OL}$ | Low-level output current | | | 8 | mA |
| $T_C$ | Operating case temperature | 0 | | 90 | °C |

### electrical characteristics over recommended ranges of supply voltage and operating case temperature (unless otherwise noted)

| PARAMETER | | TEST CONDITIONS | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| $V_{OH}$ | High-level output voltage | $DV_{DD}$ = MIN, | $I_{OH}$ = MAX | 2.4 | | | V |
| $V_{OL}$ | Low-level output voltage | $DV_{DD}$ = MIN, | $I_{OL}$ = MAX | | | 0.6 | V |
| $I_I$ | Input current‡ | $V_I = V_{SS}$ to $DV_{DD}$ | | | | ±10 | uA |
| $I_{OZ}$ | Off-state output current | $V_O = DV_{DD}$ or 0 V | | | | ±10 | uA |
| $I_{DD2V}$ | Supply current, CPU + CPU memory access§ | $CV_{DD}$ = NOM, | CPU clock = 200 MHz | | TBD | | mA |
| $I_{DD2V}$ | Supply current, peripherals¶ | $CV_{DD}$ = NOM, | CPU clock = 200 MHz | | TBD | | mA |
| $I_{DD3V}$ | Supply current, I/O pins# | $DV_{DD}$ = NOM, | CPU clock = 200 MHz | | TBD | | mA |
| $C_i$ | Input capacitance | | | | | 10 | pF |
| $C_o$ | Output capacitance | | | | | 10 | pF |

‡ TMS and TDI are not included due to internal pullups. $\overline{TRST}$ is not included due to internal pulldown.
§ Measured with average CPU activity:
50% of time: 8 instructions per cycle, 32-bit DMEM access per cycle
50% of time: 2 instructions per cycle, 16-bit DMEM access per cycle
¶ Measured with average peripheral activity:
50% of time: Timers at max rate
McBSPs at E1 rate
DMA burst transfer between DMEM and SDRAM
50% of time: Timers at max rate
McBSPs at E1 rate
DMA servicing McBSPs
# Measured with average I/O activity (30-pF load, SDCLK on):
25% of time: Reads from external SDRAM
25% of time: Writes to external SDRAM
50% of time: No activity

## PARAMETER MEASUREMENT INFORMATION




† Typical distributed load circuit capacitance

### signal transition levels

All input and output timing parameters are referenced to 1.5 V for both “0” and “1” logic levels.

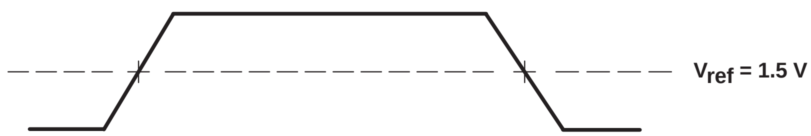



**Figure 7. Input and Output Voltage Reference Levels for ac Timing Measurements**

ADVANCE INFORMATION

## INPUT AND OUTPUT CLOCKS

### timing requirements for CLKIN† (see Figure 8)

| NO. | | | ’C6202-200 | | | | ’C6202-233 | | | | ’C6202-250 | | | | UNIT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | CLKMODE = x4 | | CLKMODE = x1 | | CLKMODE = x4 | | CLKMODE = x1 | | CLKMODE = x4 | | CLKMODE = x1 | | |
| | | | MIN | MAX | MIN | MAX | MIN | MAX | MIN | MAX | MIN | MAX | MIN | MAX | |
| 1 | $t_{c(CLKIN)}$ | Cycle time, CLKIN | 20 | | 5 | | 17.2 | | 4.3 | | 16 | | 4 | | ns |
| 2 | $t_{w(CLKINH)}$ | Pulse duration, CLKIN high | 8 | | 2.25 | | 6.9 | | 1.9 | | 6.4 | | 1.8 | | ns |
| 3 | $t_{w(CLKINL)}$ | Pulse duration, CLKIN low | 8 | | 2.25 | | 6.9 | | 1.9 | | 6.4 | | 1.8 | | ns |
| 4 | $t_{t(CLKIN)}$ | Transition time, CLKIN | | 5 | | 0.6 | | 5 | | 0.6 | | 5 | | 0.6 | ns |

† The reference points for the rise and fall transitions are measured at 20% and 80%, respectively, of $V_{IH}$.

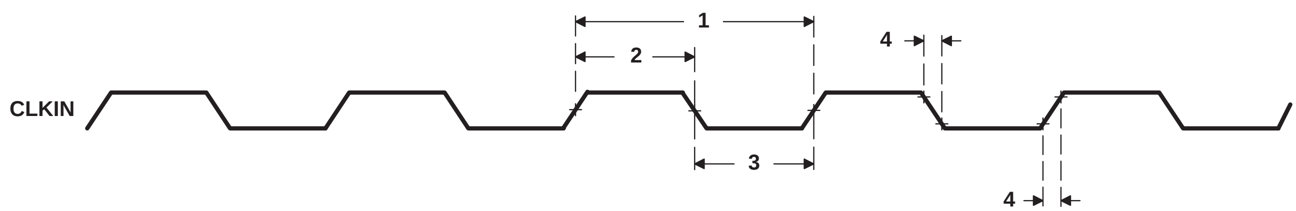



Figure 8. CLKIN Timings

### timing requirements for XCLKIN†‡ (see Figure 9)

| NO. | | | ’C6202-200 ’C6202-233 ’C6202-250 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(XCLKIN)}$ | Cycle time, XCLKIN | 4P | | ns |
| 2 | $t_{w(XCLKINH)}$ | Pulse duration, XCLKIN high | 1.8P | | ns |
| 3 | $t_{w(XCLKINL)}$ | Pulse duration, XCLKIN low | 1.8P | | ns |
| 4 | $t_{t(XCLKIN)}$ | Transition time, XCLKIN | | 0.6 | ns |

† The reference points for the rise and fall transitions are measured at 20% and 80%, respectively, of $V_{IH}$.
‡ P = 1/CPU clock frequency in nanoseconds (ns).

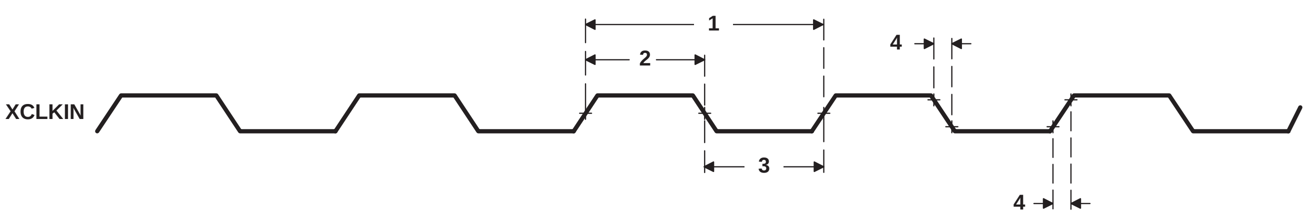



Figure 9. XCLKIN Timings

## INPUT AND OUTPUT CLOCKS (CONTINUED)

### switching characteristics for CLKOUT1†‡ (see Figure 10)

| NO. | PARAMETER | | ’C6202-200 ’C6202-233 ’C6202-250 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | CLKMODE = x4 | | CLKMODE = x1 | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{c(CKO1)}$ | Cycle time, CLKOUT1 | P – 0.7 | P + 0.7 | P – 0.7 | P + 0.7 | ns |
| 2 | $t_{w(CKO1H)}$ | Pulse duration, CLKOUT1 high | (P/2) – 0.5 | (P/2 ) + 0.5 | PH – 0.5 | PH + 0.5 | ns |
| 3 | $t_{w(CKO1L)}$ | Pulse duration, CLKOUT1 low | (P/2) – 0.5 | (P/2 ) + 0.5 | PL – 0.5 | PL + 0.5 | ns |
| 4 | $t_{t(CKO1)}$ | Transition time, CLKOUT1 | | 0.6 | | 0.6 | ns |

† PH is the high period of CLKIN in ns and PL is the low period of CLKIN in ns.
‡ P = 1/CPU clock frequency in nanoseconds (ns).

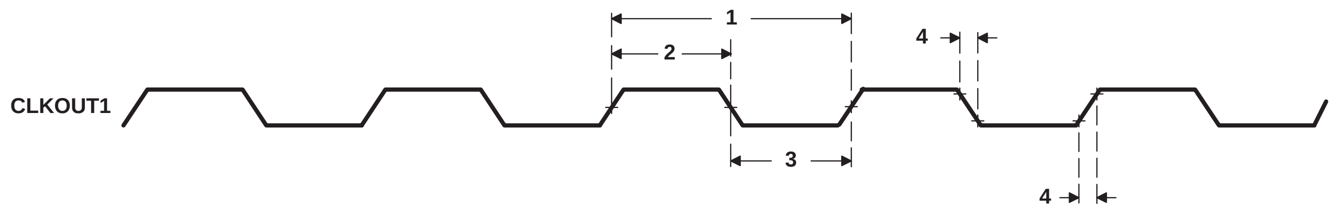



**Figure 10. CLKOUT1 Timings**

### switching characteristics for CLKOUT2‡ (see Figure 11)

| NO. | PARAMETER | | ’C6202-200 ’C6202-233 ’C6202-250 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(CKO2)}$ | Cycle time, CLKOUT2 | 2P – 0.7 | 2P + 0.7 | ns |
| 2 | $t_{w(CKO2H)}$ | Pulse duration, CLKOUT2 high | P – 0.7 | P + 0.7 | ns |
| 3 | $t_{w(CKO2L)}$ | Pulse duration, CLKOUT2 low | P – 0.7 | P + 0.7 | ns |
| 4 | $t_{t(CKO2)}$ | Transition time, CLKOUT2 | | 0.6 | ns |

‡ P = 1/CPU clock frequency in nanoseconds (ns).

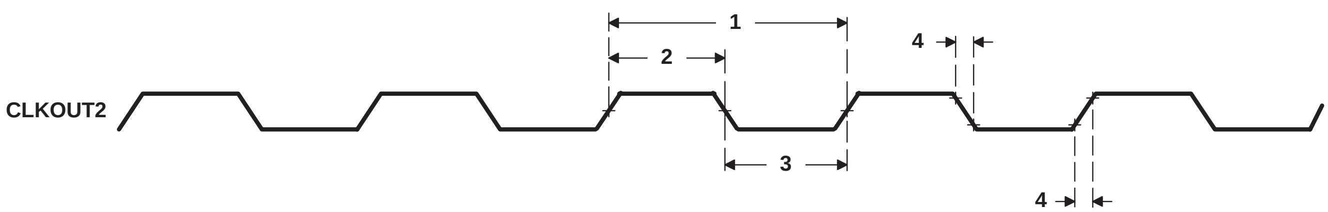



**Figure 11. CLKOUT2 Timings**

ADVANCE INFORMATION

## INPUT AND OUTPUT CLOCKS (CONTINUED)

### switching characteristics for XFCLK†‡ (see Figure 12)

| NO. | PARAMETER | | ’C6202-200 ’C6202-233 ’C6202-250 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(XFCK)}$ | Cycle time, XFCLK | D * P – 0.7 | D * P + 0.7 | ns |
| 2 | $t_{w(XFCKH)}$ | Pulse duration, XFCLK high | (D/2) * P – 0.7 | (D/2) * P + 0.7 | ns |
| 3 | $t_{w(XFCKL)}$ | Pulse duration, XFCLK low | (D/2) * P – 0.7 | (D/2) * P + 0.7 | ns |
| 4 | $t_{t(XFCK)}$ | Transition time, XFCLK | | 0.6 | ns |

† P = 1/CPU clock frequency in ns.
‡ D = 8, 6, 4, or 2; FIFO clock divide ratio, user-programmable

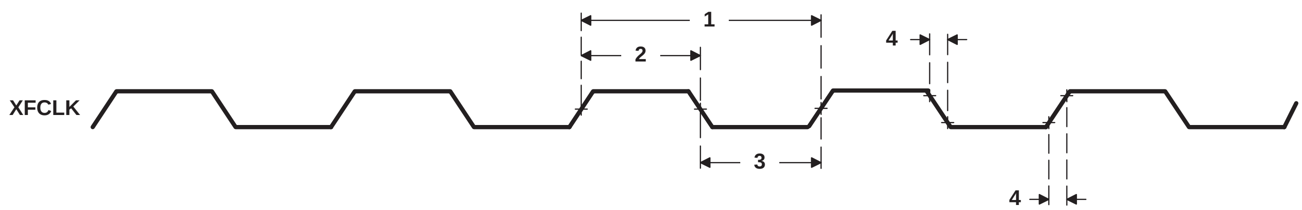



Figure 12. XFCLK Timings

## ASYNCHRONOUS MEMORY TIMING

### timing requirements for asynchronous memory cycles† (see Figure 13 – Figure 14)

| NO. | | | ’C6202-200 ’C6202-233 ’C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 6 | $t_{su(EDV-CKO1H)}$ | Setup time, read EDx valid before CLKOUT1 high | 4.0 | | ns |
| 7 | $t_{h(CKO1H-EDV)}$ | Hold time, read EDx valid after CLKOUT1 high | 0 | | ns |
| 10 | $t_{su(ARDY-CKO1H)}$ | Setup time, ARDY valid before CLKOUT1 high | 4.0 | | ns |
| 11 | $t_{h(CKO1H-ARDY)}$ | Hold time, ARDY valid after CLKOUT1 high | 0 | | ns |

† To ensure data setup time, simply program the strobe width wide enough. ARDY is internally synchronized. If ARDY does meet setup or hold time, it may be recognized in the current cycle or the next cycle. Thus, ARDY can be an asynchronous input.

### switching characteristics for asynchronous memory cycles‡ (see Figure 13 – Figure 14)

| NO. | PARAMETER | | ’C6202-200 ’C6202-233 ’C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{d(CKO1H-CEV)}$ | Delay time, CLKOUT1 high to $\overline{CEx}$ valid | 0 | 4.0 | ns |
| 2 | $t_{d(CKO1H-BEV)}$ | Delay time, CLKOUT1 high to $\overline{BEx}$ valid | 0 | 4.0 | ns |
| 3 | $t_{d(CKO1H-BEIV)}$ | Delay time, CLKOUT1 high to $\overline{BEx}$ invalid | 0 | 4.0 | ns |
| 4 | $t_{d(CKO1H-EAV)}$ | Delay time, CLKOUT1 high to EAx valid | 0 | 4.0 | ns |
| 5 | $t_{d(CKO1H-EAIV)}$ | Delay time, CLKOUT1 high to EAx invalid | 0 | 4.0 | ns |
| 8 | $t_{d(CKO1H-AOEV)}$ | Delay time, CLKOUT1 high to $\overline{AOE}$ valid | 0 | 4.0 | ns |
| 9 | $t_{d(CKO1H-AREV)}$ | Delay time, CLKOUT1 high to $\overline{ARE}$ valid | 0 | 4.0 | ns |
| 12 | $t_{d(CKO1H-EDV)}$ | Delay time, CLKOUT1 high to EDx valid | | 4.0 | ns |
| 13 | $t_{d(CKO1H-EDIV)}$ | Delay time, CLKOUT1 high to EDx invalid | 0 | | ns |
| 14 | $t_{d(CKO1H-AWEV)}$ | Delay time, CLKOUT1 high to $\overline{AWE}$ valid | 0 | 4.0 | ns |

‡ The minimum delay is also the minimum output hold after CLKOUT1 high.

ADVANCE INFORMATION

## ASYNCHRONOUS MEMORY TIMING (CONTINUED)

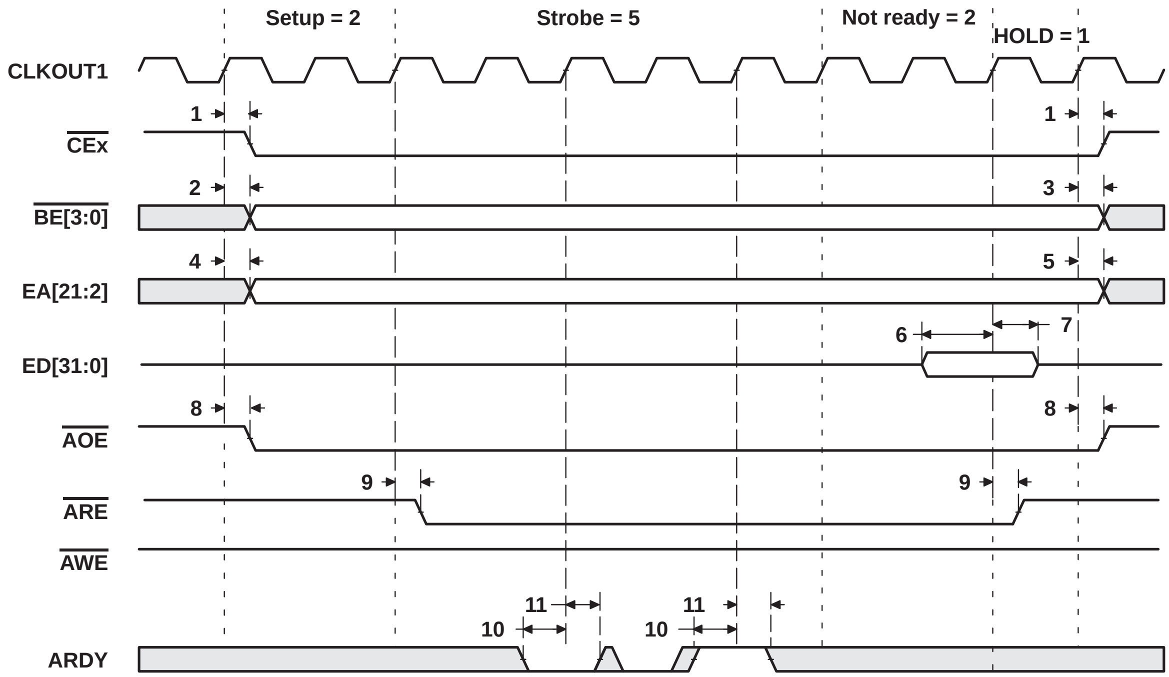



**Figure 13. Asynchronous Memory Read Timing**

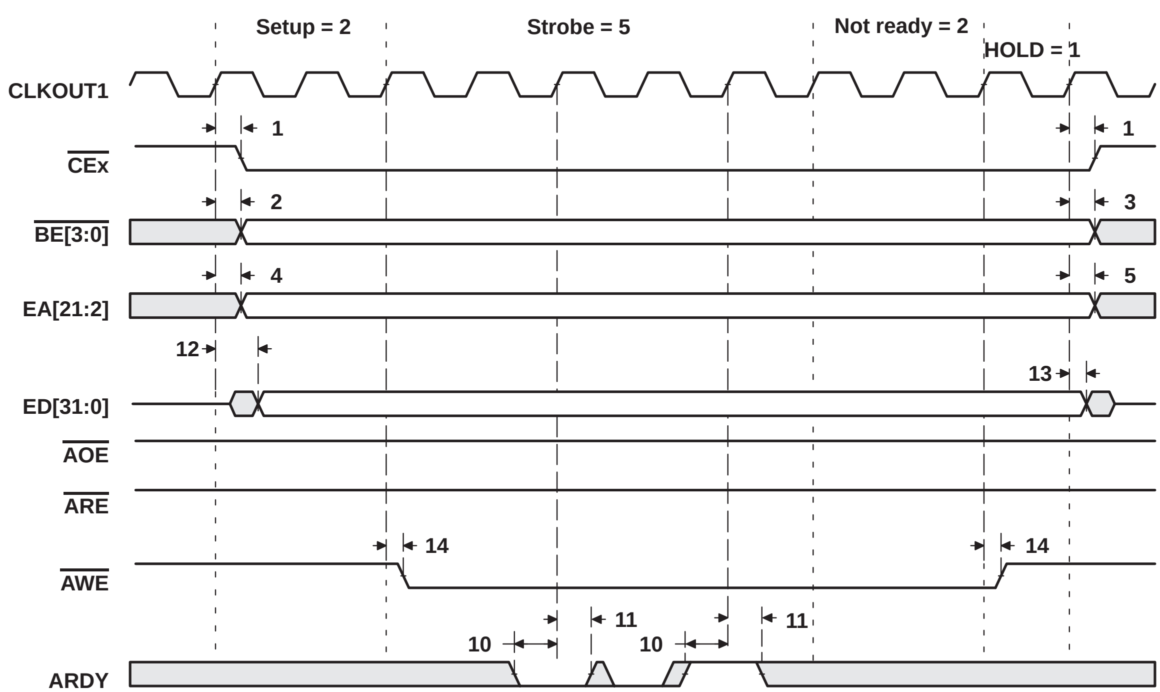



**Figure 14. Asynchronous Memory Write Timing**

## SYNCHRONOUS-BURST MEMORY TIMING

### timing requirements for synchronous-burst SRAM cycles (see Figure 15)

| NO. | | | ’C6202-200 | | ’C6202-233 | | ’C6202-250 | | UNIT |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | MIN | MAX | |
| 7 | $t_{su(EDV-CKO2H)}$ | Setup time, read EDx valid before CLKOUT2 high | 2.5 | | 2.1 | | 2 | | ns |
| 8 | $t_{h(CKO2H-EDV)}$ | Hold time, read EDx valid after CLKOUT2 high | 1.5 | | 1.5 | | 1.5 | | ns |

### switching characteristics for synchronous-burst SRAM cycles†‡ (see Figure 15 and Figure 16)

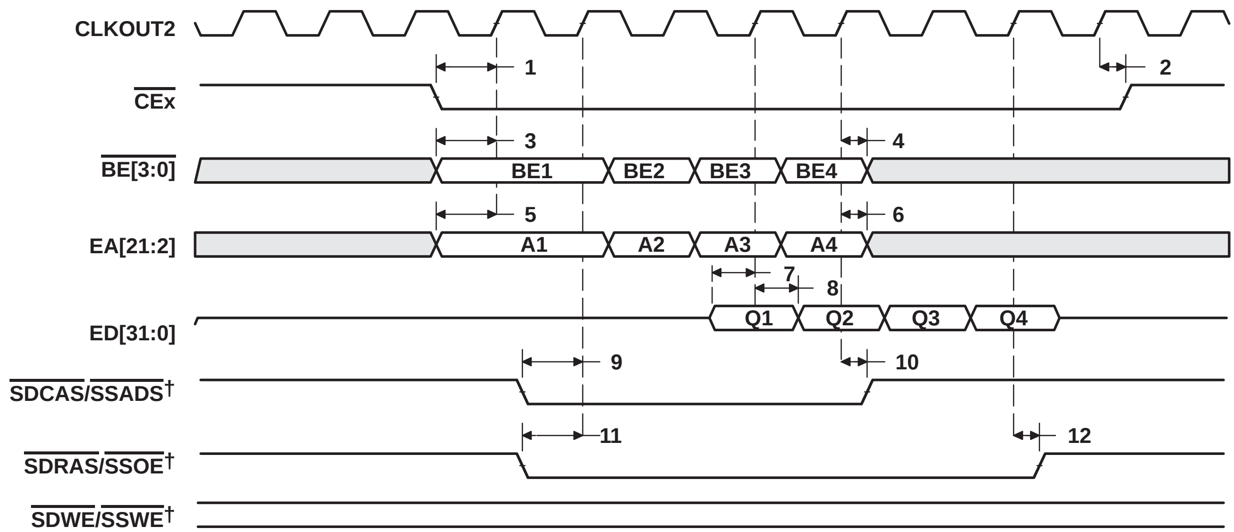
| NO. | PARAMETER | | ’C6202-200 | | ’C6202-233 | | ’C6202-250 | | UNIT |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | MIN | MAX | |
| 1 | $t_{osu(CEV-CKO2H)}$ | Output setup time, $\overline{CEx}$ valid before CLKOUT2 high | 2P – 5.5 | | 2P – 4.4 | | 2P – 3.8 | | ns |
| 2 | $t_{oh(CKO2H-CEV)}$ | Output hold time, $\overline{CEx}$ valid after CLKOUT2 high | 1 | | 1 | | 1 | | ns |
| 3 | $t_{osu(BEV-CKO2H)}$ | Output setup time, $\overline{BEx}$ valid before CLKOUT2 high | 2P – 5.5 | | 2P – 4.4 | | 2P – 3.8 | | ns |
| 4 | $t_{oh(CKO2H-BEIV)}$ | Output hold time, $\overline{BEx}$ invalid after CLKOUT2 high | 1 | | 1 | | 1 | | ns |
| 5 | $t_{osu(EAV-CKO2H)}$ | Output setup time, EAx valid before CLKOUT2 high | 2P – 5.5 | | 2P – 4.4 | | 2P – 3.8 | | ns |
| 6 | $t_{oh(CKO2H-EAIV)}$ | Output hold time, EAx invalid after CLKOUT2 high | 1 | | 1 | | 1 | | ns |
| 9 | $t_{osu(ADSV-CKO2H)}$ | Output setup time, $\overline{SDCAS}$/$\overline{SSADS}$ valid before CLKOUT2 high | 2P – 5.5 | | 2P – 4.4 | | 2P – 3.8 | | ns |
| 10 | $t_{oh(CKO2H-ADSV)}$ | Output hold time, $\overline{SDCAS}$/$\overline{SSADS}$ valid after CLKOUT2 high | 1 | | 1 | | 1 | | ns |
| 11 | $t_{osu(OEV-CKO2H)}$ | Output setup time, $\overline{SDRAS}$/$\overline{SSOE}$ valid before CLKOUT2 high | 2P – 5.5 | | 2P – 4.4 | | 2P – 3.8 | | ns |
| 12 | $t_{oh(CKO2H-OEV)}$ | Output hold time, $\overline{SDRAS}$/$\overline{SSOE}$ valid after CLKOUT2 high | 1 | | 1 | | 1 | | ns |
| 13 | $t_{osu(EDV-CKO2H)}$ | Output setup time, EDx valid before CLKOUT2 high§ | 2P – 5.5 | | 2P – 4.4 | | 2P – 3.8 | | ns |
| 14 | $t_{oh(CKO2H-EDIV)}$ | Output hold time, EDx invalid after CLKOUT2 high | 1 | | 1 | | 1 | | ns |
| 15 | $t_{osu(WEV-CKO2H)}$ | Output setup time, $\overline{SDWE}$/$\overline{SSWE}$ valid before CLKOUT2 high | 2P – 5.5 | | 2P – 4.4 | | 2P – 3.8 | | ns |
| 16 | $t_{oh(CKO2H-WEV)}$ | Output hold time, $\overline{SDWE}$/$\overline{SSWE}$ valid after CLKOUT2 high | 1 | | 1 | | 1 | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ $\overline{SDCAS}$/$\overline{SSADS}$, $\overline{SDRAS}$/$\overline{SSOE}$, and $\overline{SDWE}$/$\overline{SSWE}$ operate as $\overline{SSADS}$, $\overline{SSOE}$, and $\overline{SSWE}$, respectively, during SBSRAM accesses.
§ For the first write in a series of one or more consecutive adjacent writes, the write data is generated one CLKOUT2 cycle early to accommodate the ED enable time.

ADVANCE INFORMATION

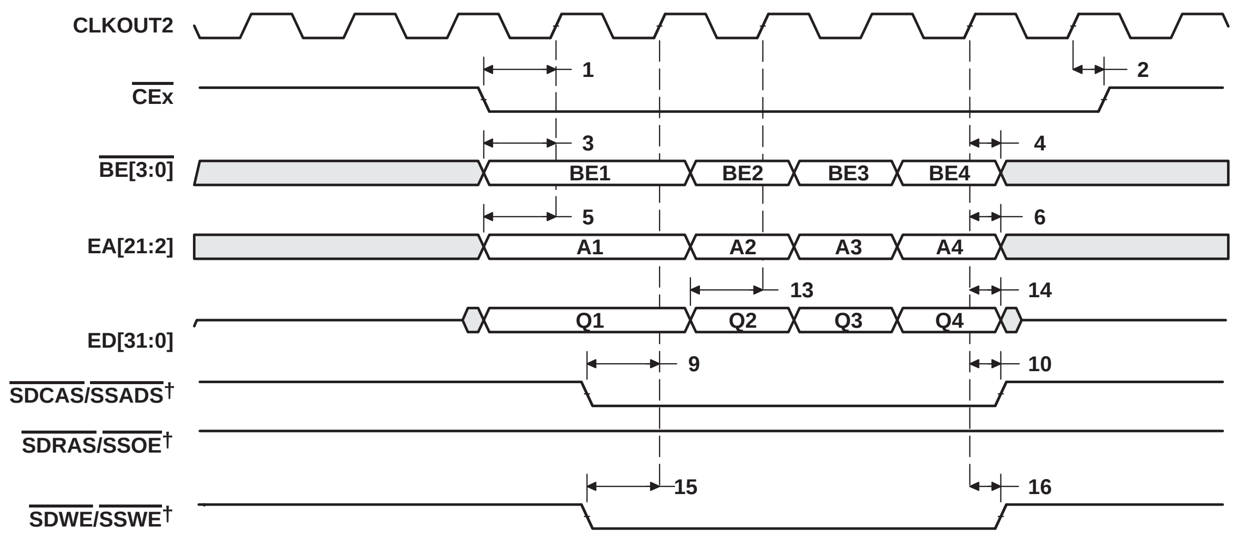
ADVANCE INFORMATION

## SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)




† SDCAS/SSADS, SDRAS/SSOE, and SDWE/SSWE operate as SSADS, SSOE, and SSWE, respectively, during SBSRAM accesses.

**Figure 15. SBSRAM Read Timing**




† SDCAS/SSADS, SDRAS/SSOE, and SDWE/SSWE operate as SSADS, SSOE, and SSWE, respectively, during SBSRAM accesses.

**Figure 16. SBSRAM Write Timing**

## SYNCHRONOUS DRAM TIMING

### timing requirements for synchronous DRAM cycles (see Figure 17)

| NO. | | | ’C6202-200 | | ’C6202-233 | | ’C6202-250 | | UNIT |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | MIN | MAX | |
| 7 | $t_{su(EDV-CKO2H)}$ | Setup time, read EDx valid before CLKOUT2 high | 1 | | 1 | | 0.5 | | ns |
| 8 | $t_{h(CKO2H-EDV)}$ | Hold time, read EDx valid after CLKOUT2 high | 3 | | 3 | | 3 | | ns |

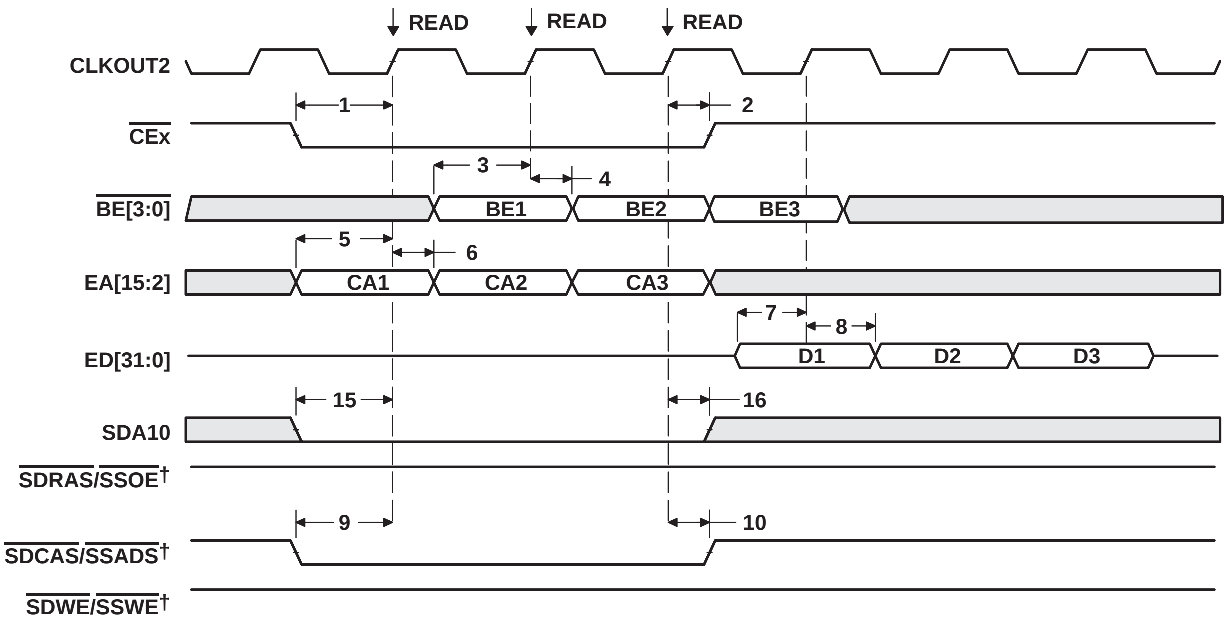
### switching characteristics for synchronous DRAM cycles†‡ (see Figure 17–Figure 22)

| NO. | PARAMETER | | ’C6202-200 | | ’C6202-233 | | ’C6202-250 | | UNIT |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | MIN | MAX | |
| 1 | $t_{osu(CEV-CKO2H)}$ | Output setup time, $\overline{\text{CEx}}$ valid before CLKOUT2 high | 2P – 6 | | 2P – 4.6 | | 2P – 4 | | ns |
| 2 | $t_{oh(CKO2H-CEV)}$ | Output hold time, $\overline{\text{CEx}}$ valid after CLKOUT2 high | 1.5 | | 1.5 | | 1.5 | | ns |
| 3 | $t_{osu(BEV-CKO2H)}$ | Output setup time, $\overline{\text{BEx}}$ valid before CLKOUT2 high | 2P – 6 | | 2P – 4.6 | | 2P – 4 | | ns |
| 4 | $t_{oh(CKO2H-BEIV)}$ | Output hold time, $\overline{\text{BEx}}$ invalid after CLKOUT2 high | 1.5 | | 1.5 | | 1.5 | | ns |
| 5 | $t_{osu(EAV-CKO2H)}$ | Output setup time, EAx valid before CLKOUT2 high | 2P – 6 | | 2P – 4.6 | | 2P – 4 | | ns |
| 6 | $t_{oh(CKO2H-EAIV)}$ | Output hold time, EAx invalid after CLKOUT2 high | 1.5 | | 1.5 | | 1.5 | | ns |
| 9 | $t_{osu(CASV-CKO2H)}$ | Output setup time, $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$ valid before CLKOUT2 high | 2P – 6 | | 2P – 4.6 | | 2P – 4 | | ns |
| 10 | $t_{oh(CKO2H-CASV)}$ | Output hold time, $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$ valid after CLKOUT2 high | 1.5 | | 1.5 | | 1.5 | | ns |
| 11 | $t_{osu(EDV-CKO2H)}$ | Output setup time, EDx valid before CLKOUT2 high§ | 2P – 6 | | 2P – 4.6 | | 2P – 4 | | ns |
| 12 | $t_{oh(CKO2H-EDIV)}$ | Output hold time, EDx invalid after CLKOUT2 high | 1.5 | | 1.5 | | 1.5 | | ns |
| 13 | $t_{osu(WEV-CKO2H)}$ | Output setup time, $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ valid before CLKOUT2 high | 2P – 6 | | 2P – 4.6 | | 2P – 4 | | ns |
| 14 | $t_{oh(CKO2H-WEV)}$ | Output hold time, $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ valid after CLKOUT2 high | 1.5 | | 1.5 | | 1.5 | | ns |
| 15 | $t_{osu(SDA10V-CKO2H)}$ | Output setup time, SDA10 valid before CLKOUT2 high | 2P – 6 | | 2P – 4.6 | | 2P – 4 | | ns |
| 16 | $t_{oh(CKO2H-SDA10IV)}$ | Output hold time, SDA10 invalid after CLKOUT2 high | 1.5 | | 1.5 | | 1.5 | | ns |
| 17 | $t_{osu(RASV-CKO2H)}$ | Output setup time, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$ valid before CLKOUT2 high | 2P – 6 | | 2P – 4.6 | | 2P – 4 | | ns |
| 18 | $t_{oh(CKO2H-RASV)}$ | Output hold time, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$ valid after CLKOUT2 high | 1.5 | | 1.5 | | 1.5 | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$, and $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ operate as $\overline{\text{SDCAS}}$, $\overline{\text{SDRAS}}$, and $\overline{\text{SDWE}}$, respectively, during SDRAM accesses.
§ For the first write in a series of one or more consecutive adjacent writes, the write data is generated one CLKOUT2 cycle early to accommodate the ED enable time.

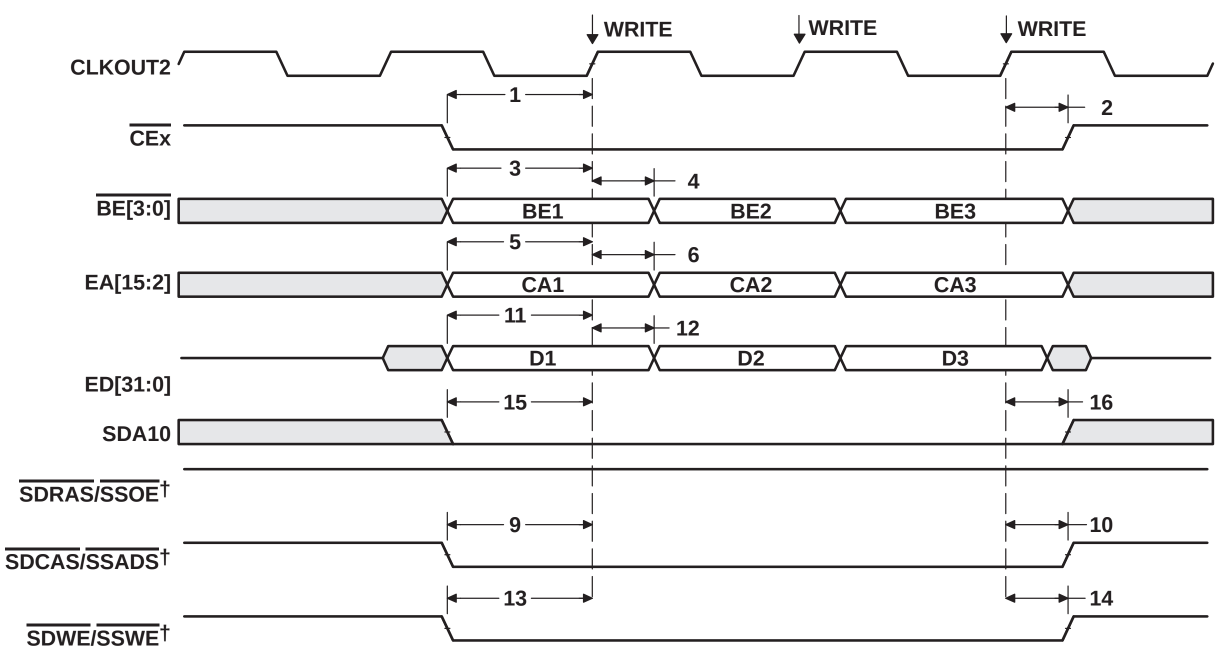
ADVANCE INFORMATION

## SYNCHRONOUS DRAM TIMING (CONTINUED)




† SDCAS/SSADS, SDRAS/SSOE, and SDWE/SSWE operate as SDCAS, SDRAS, and SDWE, respectively, during SDRAM accesses.

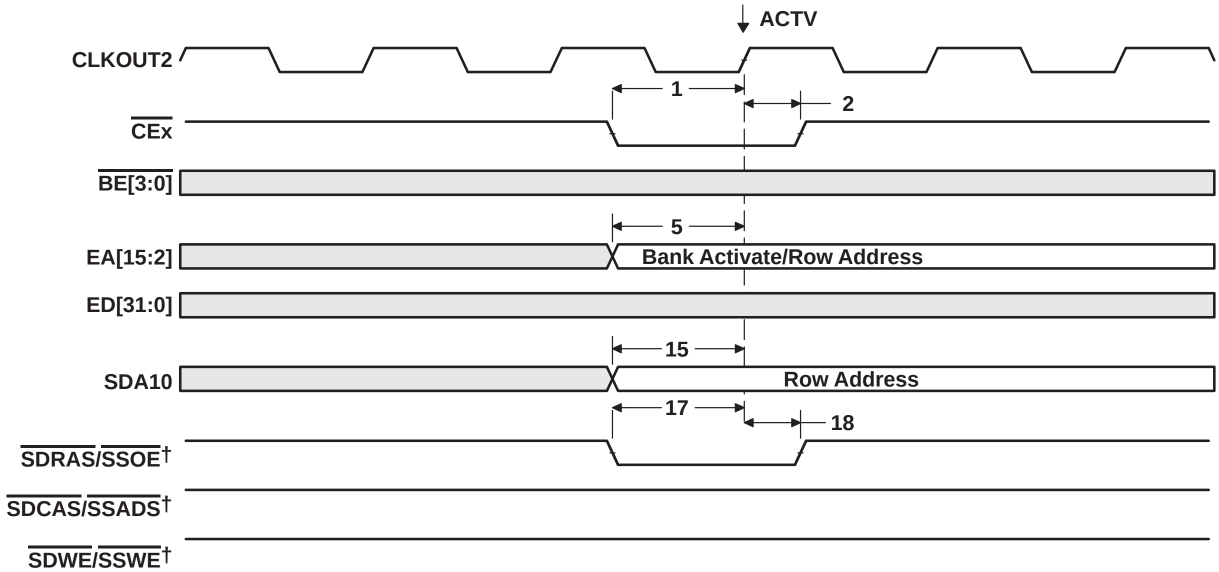
**Figure 17. Three SDRAM READ Commands**




† SDCAS/SSADS, SDRAS/SSOE, and SDWE/SSWE operate as SDCAS, SDRAS, and SDWE, respectively, during SDRAM accesses.

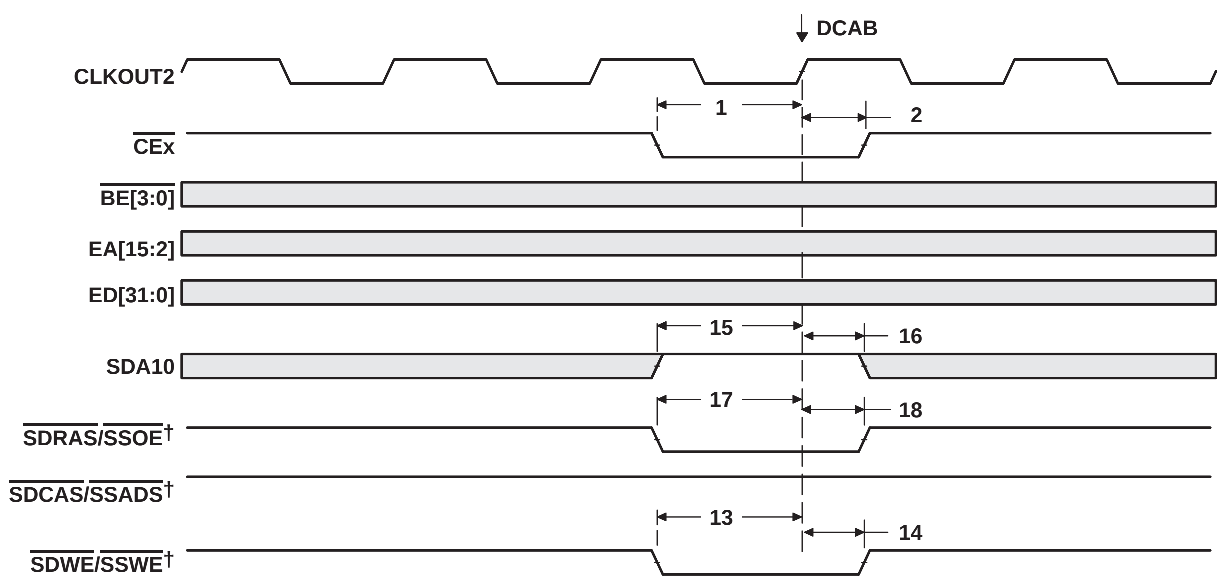
**Figure 18. Three SDRAM WRT Commands**

## SYNCHRONOUS DRAM TIMING (CONTINUED)




† $\overline{\text{SDCAS}}/\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}/\overline{\text{SSOE}}$, and $\overline{\text{SDWE}}/\overline{\text{SSWE}}$ operate as $\overline{\text{SDCAS}}$, $\overline{\text{SDRAS}}$, and $\overline{\text{SDWE}}$, respectively, during SDRAM accesses.

**Figure 19. SDRAM ACTV Command**



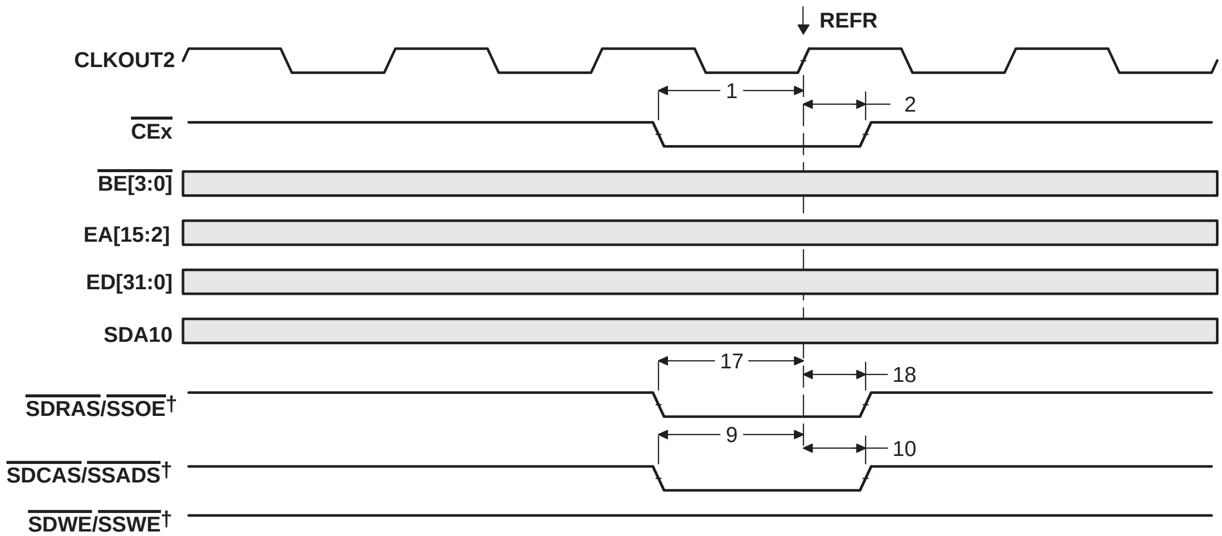

† $\overline{\text{SDCAS}}/\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}/\overline{\text{SSOE}}$, and $\overline{\text{SDWE}}/\overline{\text{SSWE}}$ operate as $\overline{\text{SDCAS}}$, $\overline{\text{SDRAS}}$, and $\overline{\text{SDWE}}$, respectively, during SDRAM accesses.

**Figure 20. SDRAM DCAB Command**

ADVANCE INFORMATION

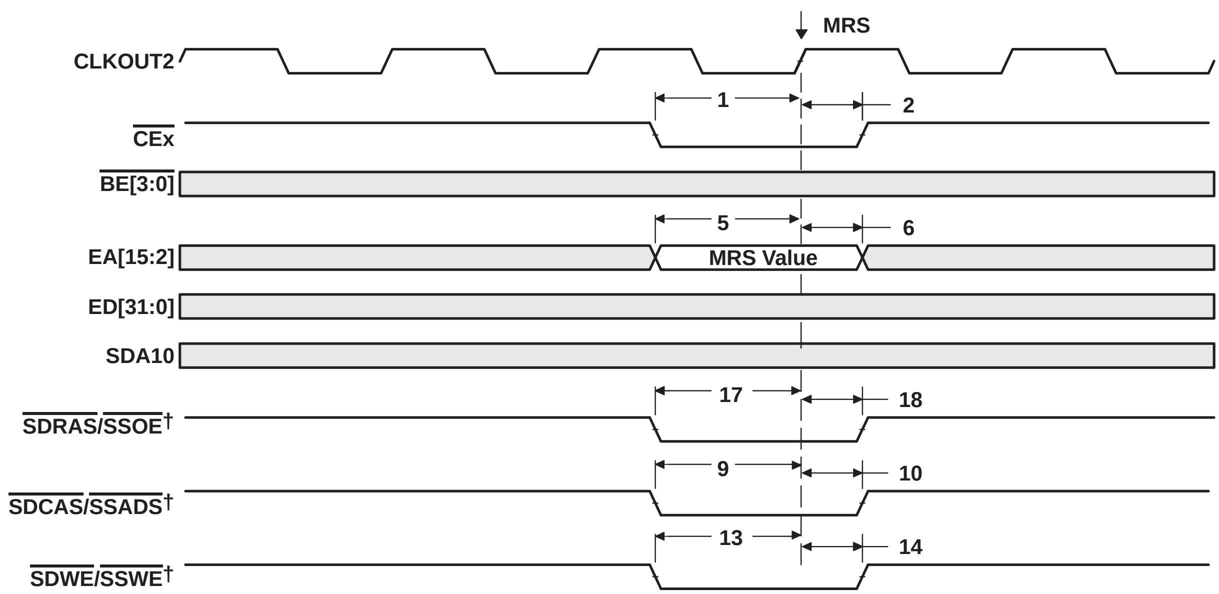
ADVANCE INFORMATION

## SYNCHRONOUS DRAM TIMING (CONTINUED)




† SDCAS/SSADS, SDRAS/SSOE, and SDWE/SSWE operate as SDCAS, SDRAS, and SDWE, respectively, during SDRAM accesses.

**Figure 21. SDRAM REFR Command**




† SDCAS/SSADS, SDRAS/SSOE, and SDWE/SSWE operate as SDCAS, SDRAS, and SDWE, respectively, during SDRAM accesses.

**Figure 22. SDRAM MRS Command**

## $\overline{\text{HOLD}}$/$\overline{\text{HOLDA}}$ TIMING

### timing requirements for the $\overline{\text{HOLD}}$/$\overline{\text{HOLDA}}$ cycles† (see Figure 23)

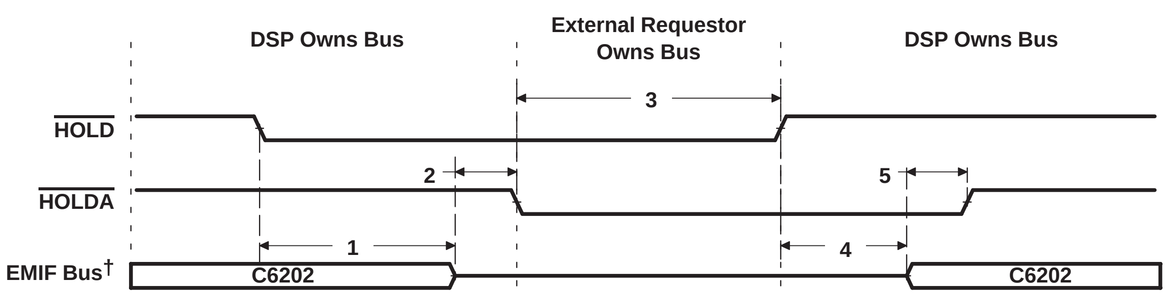
| NO. | | | ’C6202-200 ’C6202-233 ’C6202-250 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 3 | $t_{oh(HOLDAL-HOLD)}$ | Hold time, $\overline{\text{HOLD}}$ low after $\overline{\text{HOLDA}}$ low | P | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

### switching characteristics for the $\overline{\text{HOLD}}$/$\overline{\text{HOLDA}}$ cycles†‡ (see Figure 23)

| NO. | PARAMETER | | ’C6202-200 ’C6202-233 ’C6202-250 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{R(HOLDL-EMHZ)}$ | Response time, $\overline{\text{HOLD}}$ low to EMIF Bus high impedance | 4P | § | ns |
| 2 | $t_{d(EMHZ-HOLDAL)}$ | Delay time, EMIF Bus high impedance to $\overline{\text{HOLDA}}$ low | 0 | 2P | ns |
| 4 | $t_{R(HOLDH-EMLZ)}$ | Response time, $\overline{\text{HOLD}}$ high to EMIF Bus low impedance | 3P | 7P | ns |
| 5 | $t_{d(EMLZ-HOLDAH)}$ | Delay time, EMIF Bus low impedance to $\overline{\text{HOLDA}}$ high | 0 | 2P | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ EMIF Bus consists of $\overline{\text{CE[3:0]}}$, $\overline{\text{BE[3:0]}}$, ED[31:0], EA[21:2], $\overline{\text{ARE}}$, $\overline{\text{AOE}}$, $\overline{\text{AWE}}$, $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$, $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$, and SDA10.
§ All pending EMIF transactions are allowed to complete before $\overline{\text{HOLDA}}$ is asserted. The worst case for this is an asynchronous read or write with external ARDY used or a minimum of eight consecutive SDRAM reads or writes when RBTR8 = 1. If no bus transactions are occurring, then the minimum delay time can be achieved. Also, bus hold can be indefinitely delayed by setting NOHOLD = 1.




† EMIF Bus consists of $\overline{\text{CE[3:0]}}$, $\overline{\text{BE[3:0]}}$, ED[31:0], EA[21:2], $\overline{\text{ARE}}$, $\overline{\text{AOE}}$, $\overline{\text{AWE}}$, $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$, $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$, and SDA10.

**Figure 23. $\overline{\text{HOLD}}$/$\overline{\text{HOLDA}}$ Timing**

ADVANCE INFORMATION

ADVANCE INFORMATION

## RESET TIMING

### timing requirements for reset (see Figure 24)

| NO. | | | ’C6202-200 ’C6202-233 ’C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{w(RST)}$ | Width of the $\overline{\text{RESET}}$ pulse (PLL stable)† | 10 | | CLKOUT1 cycles |
| | | Width of the $\overline{\text{RESET}}$ pulse (PLL needs to sync up)‡ | 250 | | µs |
| 11 | $t_{su(XD)}$ | Setup time, XD configuration bits valid before $\overline{\text{RESET}}$ high§ | 5 | | CLKOUT1 cycles |
| 12 | $t_{h(XD)}$ | Hold time, XD configuration bits valid after $\overline{\text{RESET}}$ high§ | 5 | | CLKOUT1 cycles |

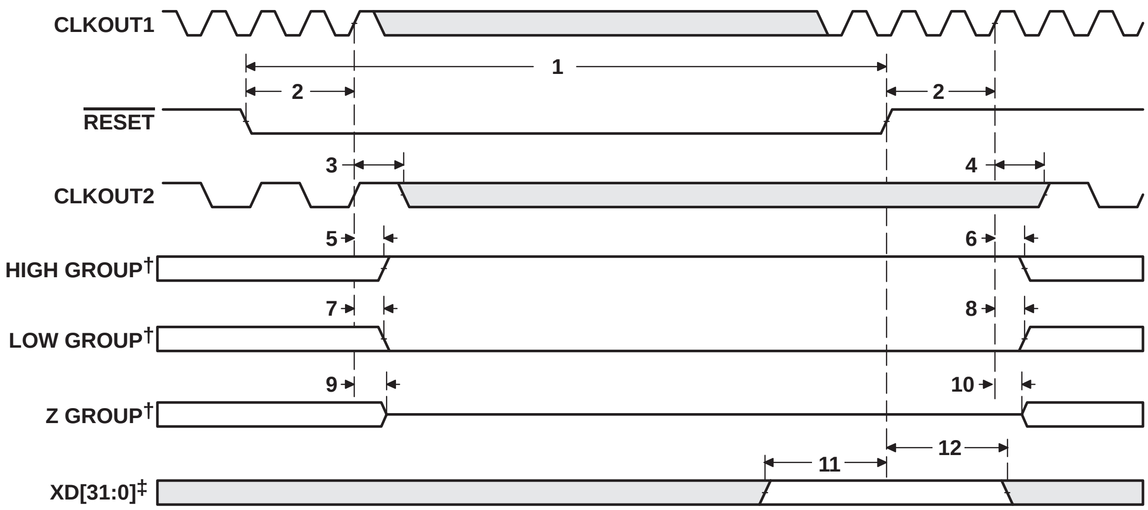
† This parameter applies to CLKMODE x1 when CLKIN is stable and applies to CLKMODE x4 when CLKIN and PLL are stable.
‡ This parameter only applies to CLKMODE x4. The $\overline{\text{RESET}}$ signal is not connected internally to the clock PLL circuit. The PLL, however, may need up to 250 µs to stabilize following device power up or after PLL configuration has been changed. During that time, $\overline{\text{RESET}}$ must be asserted to ensure proper device operation. See the *clock PLL* section for PLL lock times.
§ XD[31:0] are the boot configuration pins during device reset.

### switching characteristics during reset¶ (see Figure 24)

| NO. | PARAMETER | | ’C6202-200 ’C6202-233 ’C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 2 | $t_{R(RST)}$ | Response time to change of value in $\overline{\text{RESET}}$ signal | 2 | | CLKOUT1 cycles |
| 3 | $t_{d(CKO1H-CKO2IV)}$ | Delay time, CLKOUT1 high to CLKOUT2 invalid | –1 | 10 | ns |
| 4 | $t_{d(CKO1H-CKO2V)}$ | Delay time, CLKOUT1 high to CLKOUT2 valid | –1 | 10 | ns |
| 5 | $t_{d(CKO1H-XFCKIV)}$ | Delay time, CLKOUT1 high to high group invalid | –1 | 10 | ns |
| 6 | $t_{d(CKO1H-XFCKV)}$ | Delay time, CLKOUT1 high to high group valid | –1 | 10 | ns |
| 7 | $t_{d(CKO1H-LOWIV)}$ | Delay time, CLKOUT1 high to low group invalid | –1 | 10 | ns |
| 8 | $t_{d(CKO1H-LOWV)}$ | Delay time, CLKOUT1 high to low group valid | –1 | 10 | ns |
| 9 | $t_{d(CKO1H-ZHZ)}$ | Delay time, CLKOUT1 high to Z group high impedance | –1 | 10 | ns |
| 10 | $t_{d(CKO1H-ZV)}$ | Delay time, CLKOUT1 high to Z group valid | –1 | 10 | ns |

¶ High group consists of: XFCLK
Low group consists of: IACK, INUM[3:0], DMAC[3:0], PD, TOUT0, and TOUT1
Z group consists of: EA[21:2], ED[31:0], $\overline{\text{CE}}$[3:0], $\overline{\text{BE}}$[3:0], $\overline{\text{ARE}}$, $\overline{\text{AWE}}$, $\overline{\text{AOE}}$, $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$, $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$, SDA10, CLKX0, CLKX1, CLKX2, FSX0, FSX1, FSX2, DX0, DX1, DX2, CLKR0, CLKR1, CLKR2, FSR0, FSR1, FSR2, $\overline{\text{XCE}}$[3:0], $\overline{\text{XBE}}$[3:0]/XA[5:2], $\overline{\text{XOE}}$, $\overline{\text{XRE}}$, $\overline{\text{XWE}}$/XWAIT, $\overline{\text{XAS}}$, XW/R, XRDY, XBLAST, XHOLD, and XHOLDA

## RESET TIMING (CONTINUED)




† High group consists of: XFCLK
Low group consists of: IACK, INUM[3:0], DMAC[3:0], PD, TOUT0, and TOUT1.
Z group consists of: EA[21:2], ED[31:0], CE[3:0], BE[3:0], ARE, AWE, AOE, SDCAS/SSADS, SDRAS/SSOE, SDWE/SSWE, SDA10, CLKX0, CLKX1, CLKX2, FSX0, FSX1, FSX2, DX0, DX1, DX2, CLKR0, CLKR1, CLKR2, FSR0, FSR1, FSR2, XCE[3:0], XBE[3:0]/XA[5:2], XOE, XRE, XWE/XWAIT, XAS, XW/R, XRDY, XBLAST, XHOLD, and XHOLDA.

‡ XD[31:0] are the boot configuration pins during device reset.

**Figure 24. Reset Timing**

ADVANCE INFORMATION

## EXTERNAL INTERRUPT TIMING

### timing requirements for interrupt response cycles† (see Figure 25)

| NO. | | | ’C6202-200 ’C6202-233 ’C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 2 | $t_{w(ILOW)}$ | Width of the interrupt pulse low | 2P | | ns |
| 3 | $t_{w(IHIGH)}$ | Width of the interrupt pulse high | 2P | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

### switching characteristics during interrupt response cycles† (see Figure 25)

| NO. | PARAMETER | | ’C6202-200 ’C6202-233 ’C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{R(EINTH – IACKH)}$ | Response time, EXT_INTx high to IACK high | 9P | | ns |
| 4 | $t_{d(CKO2L-IACKV)}$ | Delay time, CLKOUT2 low to IACK valid | 0 | 10 | ns |
| 5 | $t_{d(CKO2L-INUMV)}$ | Delay time, CLKOUT2 low to INUMx valid | 0 | 10 | ns |
| 6 | $t_{d(CKO2L-INUMIV)}$ | Delay time, CLKOUT2 low to INUMx invalid | 0 | 10 | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

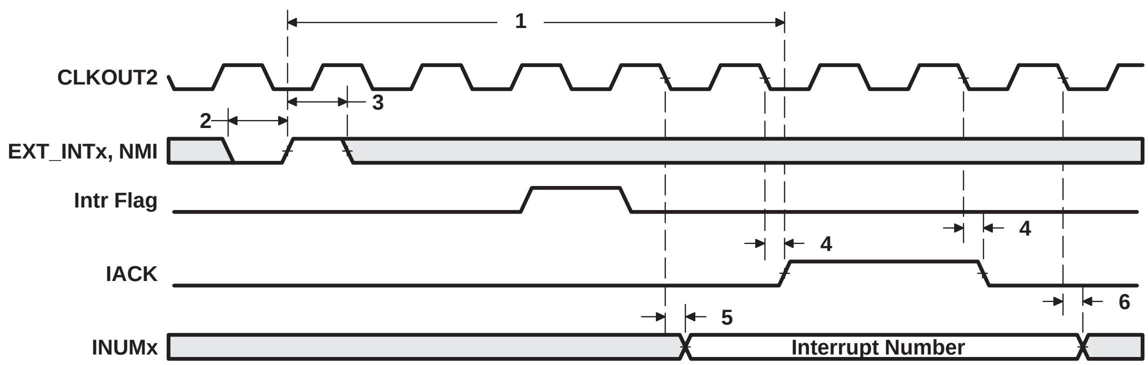



Figure 25. Interrupt Timing

## EXPANSION BUS SYNCHRONOUS FIFO TIMING

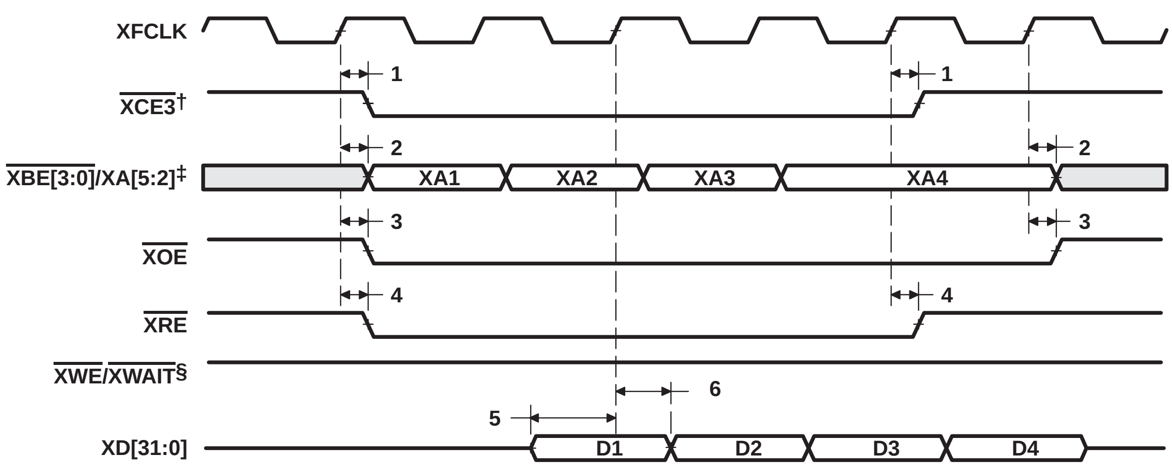
### timing requirements for synchronous FIFO interface (see Figure 26, Figure 27, and Figure 28)

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 5 | $t_{su(XDV\text{-}XFCKH)}$ | Setup time, read XDx valid before XFCLK high | 2.5 | | ns |
| 6 | $t_{h(XFCKH\text{-}XDV)}$ | Hold time, read XDx valid after XFCLK high | 2 | | ns |

### switching characteristics for synchronous FIFO interface (see Figure 26, Figure 27, and Figure 28)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{d(XFCKH\text{-}XCEV)}$ | Delay time, XFCLK high to $\overline{\text{XCE}}$x valid | 1.5 | 5.2 | ns |
| 2 | $t_{d(XFCKH\text{-}XAV)}$ | Delay time, XFCLK high to $\overline{\text{XBE[3:0]}}$/XA[5:2] valid† | 1.5 | 5.2 | ns |
| 3 | $t_{d(XFCKH\text{-}XOEV)}$ | Delay time, XFCLK high to $\overline{\text{XOE}}$ valid | 1.5 | 5.2 | ns |
| 4 | $t_{d(XFCKH\text{-}XREV)}$ | Delay time, XFCLK high to $\overline{\text{XRE}}$ valid | 1.5 | 5.2 | ns |
| 7 | $t_{d(XFCKH\text{-}XWEV)}$ | Delay time, XFCLK high to $\overline{\text{XWE}}$/$\overline{\text{XWAIT}}$‡ valid | 1.5 | 5.2 | ns |
| 8 | $t_{d(XFCKH\text{-}XDV)}$ | Delay time, XFCLK high to XDx valid | | 5.2 | ns |
| 9 | $t_{d(XFCKH\text{-}XDIV)}$ | Delay time, XFCLK high to XDx invalid | 1.5 | | ns |

† $\overline{\text{XBE[3:0]}}$/XA[5:2] operates as address signals XA[5:2] during synchronous FIFO accesses.
‡ XWE/XWAIT operates as the write enable signal $\overline{\text{XWE}}$ during synchronous FIFO accesses.



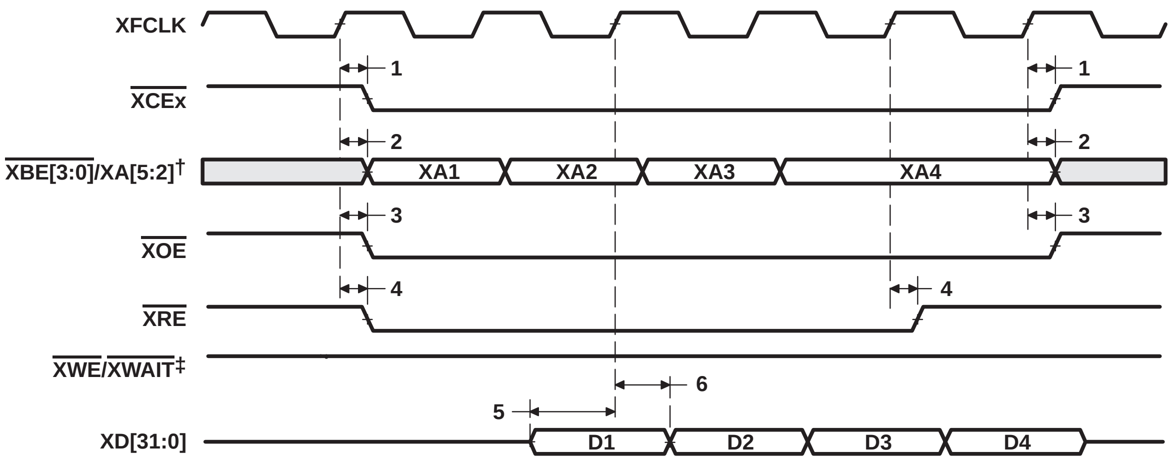

† FIFO read (glueless) mode only available in $\overline{\text{XCE3}}$.
‡ $\overline{\text{XBE[3:0]}}$/XA[5:2] operates as address signals XA[5:2] during synchronous FIFO accesses.
§ XWE/XWAIT operates as the write enable signal $\overline{\text{XWE}}$ during synchronous FIFO accesses.

**Figure 26. FIFO Read Timing (Glueless Read Mode)**

ADVANCE INFORMATION

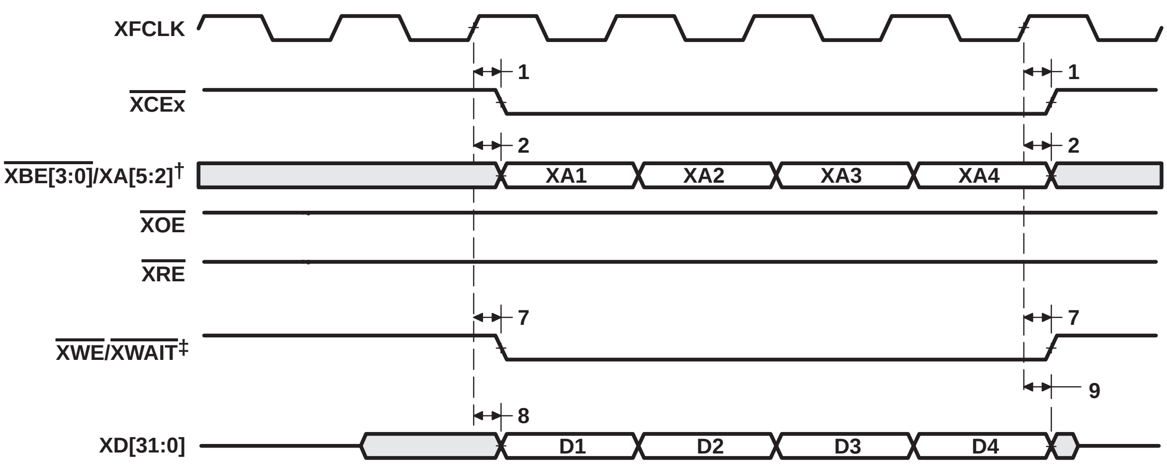
ADVANCE INFORMATION

## EXPANSION BUS SYNCHRONOUS FIFO TIMING (CONTINUED)




† XBE[3:0]/XA[5:2] operates as address signals XA[5:2] during synchronous FIFO accesses.
‡ XWE/XWAIT operates as the write enable signal XWE during synchronous FIFO accesses.

**Figure 27. FIFO Read Timing**




† XBE[3:0]/XA[5:2] operates as address signals XA[5:2] during synchronous FIFO accesses.
‡ XWE/XWAIT operates as the write enable signal XWE during synchronous FIFO accesses.

**Figure 28. FIFO Write Timing**

## EXPANSION BUS ASYNCHRONOUS PERIPHERAL TIMING

### timing requirements for asynchronous peripheral cycles† (see Figure 29–Figure 30)

| NO. | | | 'C6202-200 'C6202-233 'C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 4 | $t_{su(XDV-CKO1H)}$ | Setup time, read XDx valid before CLKOUT1 high | 4.0 | | ns |
| 5 | $t_{h(CKO1H-XDV)}$ | Hold time, read XDx valid after CLKOUT1 high | 0 | | ns |
| 8 | $t_{su(XRY-CKO1H)}$ | Setup time, XRDY valid before CLKOUT1 high | 4.0 | | ns |
| 9 | $t_{h(CKO1H-XRY)}$ | Hold time, XRDY valid after CLKOUT1 high | 0 | | ns |

† To ensure data setup time, simply program the strobe width wide enough. XRDY is internally synchronized. If XRDY does meet setup or hold time, it may be recognized in the current cycle or the next cycle. Thus, XRDY can be an asynchronous input.

### switching characteristics for asynchronous peripheral cycles‡§¶ (see Figure 29–Figure 30)

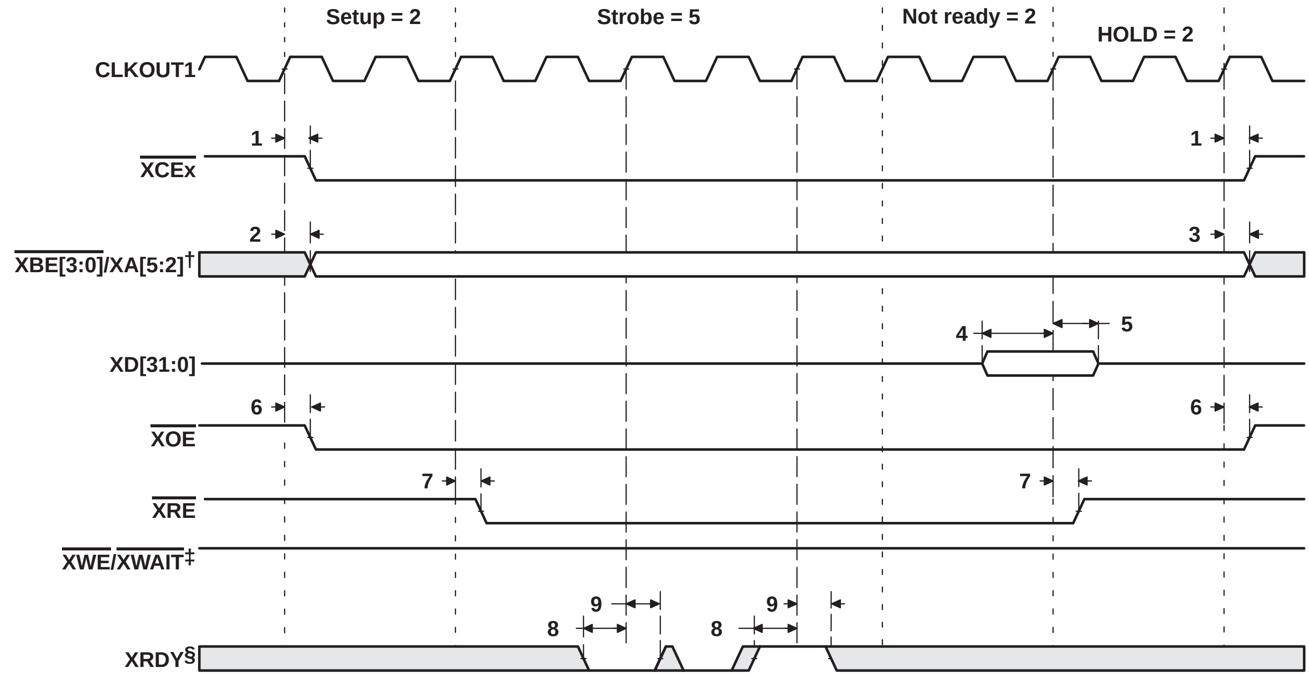
| NO. | PARAMETER | | 'C6202-200 'C6202-233 'C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{d(CKO1H-XCEV)}$ | Delay time, CLKOUT1 high to $\overline{\text{XCEx}}$ valid | 0 | 4.0 | ns |
| 2 | $t_{d(CKO1H-XAV)}$ | Delay time, CLKOUT1 high to $\overline{\text{XBE[3:0]}}$/XA[5:2] valid | 0 | 4.0 | ns |
| 3 | $t_{d(CKO1H-XAIV)}$ | Delay time, CLKOUT1 high to $\overline{\text{XBE[3:0]}}$/XA[5:2] invalid | 0 | 4.0 | ns |
| 6 | $t_{d(CKO1H-XOEV)}$ | Delay time, CLKOUT1 high to $\overline{\text{XOE}}$ valid | 0 | 4.0 | ns |
| 7 | $t_{d(CKO1H-XREV)}$ | Delay time, CLKOUT1 high to $\overline{\text{XRE}}$ valid | 0 | 4.0 | ns |
| 10 | $t_{d(CKO1H-XDV)}$ | Delay time, CLKOUT1 high to XDx valid | | 4.0 | ns |
| 11 | $t_{d(CKO1H-XDIV)}$ | Delay time, CLKOUT1 high to XDx invalid | 0 | | ns |
| 12 | $t_{d(CKO1H-XWEV)}$ | Delay time, CLKOUT1 high to $\overline{\text{XWE}}$/$\overline{\text{XWAIT}}$ valid | 0 | 4.0 | ns |

‡ The minimum delay is also the minimum output hold after CLKOUT1 high.
§ $\overline{\text{XBE[3:0]}}$/XA[5:2] operates as address signals XA[5:2] during asynchronous peripheral accesses.
¶ $\overline{\text{XWE}}$/XWAIT operates as the write enable signal $\overline{\text{XWE}}$ during asynchronous peripheral accesses.

ADVANCE INFORMATION

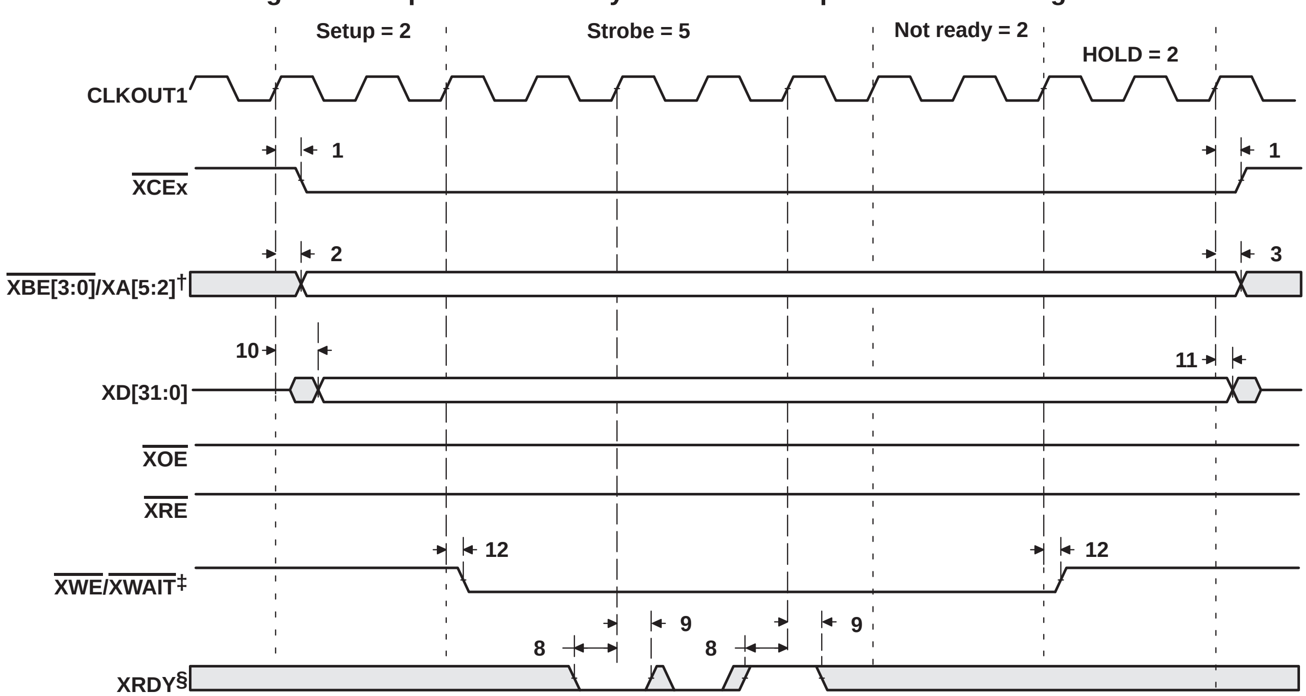
ADVANCE INFORMATION

## EXPANSION BUS ASYNCHRONOUS PERIPHERAL TIMING (CONTINUED)




† XBE[3:0]/XA[5:2] operates as address signals XA[5:2] during asynchronous peripheral accesses.
‡ XWE/XWAIT operates as the write enable signal XWE during asynchronous peripheral accesses.
§ XRDY operates as active-high ready input during asynchronous peripheral accesses.

**Figure 29. Expansion Bus Asynchronous Peripheral Read Timing**




† XBE[3:0]/XA[5:2] operates as address signals XA[5:2] during asynchronous peripheral accesses.
‡ XWE/XWAIT operates as the write enable signal XWE during asynchronous peripheral accesses.
§ XRDY operates as active-high ready input during asynchronous peripheral accesses.

**Figure 30. Expansion Bus Asynchronous Peripheral Write Timing**

## EXPANSION BUS SYNCHRONOUS HOST PORT TIMING

### timing requirements with external device as bus master (see Figure 31 and Figure 32)

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{su(XCSV-XCKIH)}$ | Setup time, $\overline{XCS}$ valid before XCLKIN high | 4 | | ns |
| 2 | $t_{h(XCKIH-XCS)}$ | Hold time, $\overline{XCS}$ valid after XCLKIN high | 2.3 | | ns |
| 3 | $t_{su(XAS-XCKIH)}$ | Setup time, $\overline{XAS}$ valid before XCLKIN high | 4 | | ns |
| 4 | $t_{h(XCKIH-XAS)}$ | Hold time, $\overline{XAS}$ valid after XCLKIN high | 2.3 | | ns |
| 5 | $t_{su(XCTL-XCKIH)}$ | Setup time, XCNTL valid before XCLKIN high | 4 | | ns |
| 6 | $t_{h(XCKIH-XCTL)}$ | Hold time, XCNTL valid after XCLKIN high | 2.3 | | ns |
| 7 | $t_{su(XWR-XCKIH)}$ | Setup time, XW/R valid before XCLKIN high† | 4 | | ns |
| 8 | $t_{h(XCKIH-XWR)}$ | Hold time, XW/R valid after XCLKIN high† | 2.3 | | ns |
| 9 | $t_{su(XBLTV-XCKIH)}$ | Setup time, XBLAST valid before XCLKIN high‡ | 4 | | ns |
| 10 | $t_{h(XCKIH-XBLTV)}$ | Hold time, XBLAST valid after XCLKIN high‡ | 2.3 | | ns |
| 16 | $t_{su(XBEV-XCKIH)}$ | Setup time, $\overline{XBE[3:0]}$/XA[5:2] valid before XCLKIN high§ | 4 | | ns |
| 17 | $t_{h(XCKIH-XBEV)}$ | Hold time, $\overline{XBE[3:0]}$/XA[5:2] valid after XCLKIN high§ | 2.3 | | ns |
| 18 | $t_{su(XD-XCKIH)}$ | Setup time, XDx valid before XCLKIN high | 4 | | ns |
| 19 | $t_{h(XCKIH-XD)}$ | Hold time, XDx valid after XCLKIN high | 2.3 | | ns |

† XW/R input/output polarity selected at boot.
‡ XBLAST input polarity selected at boot.
§ $\overline{XBE[3:0]}$/XA[5:2] operates as byte enables $\overline{XBE[3:0]}$ during host-port accesses.

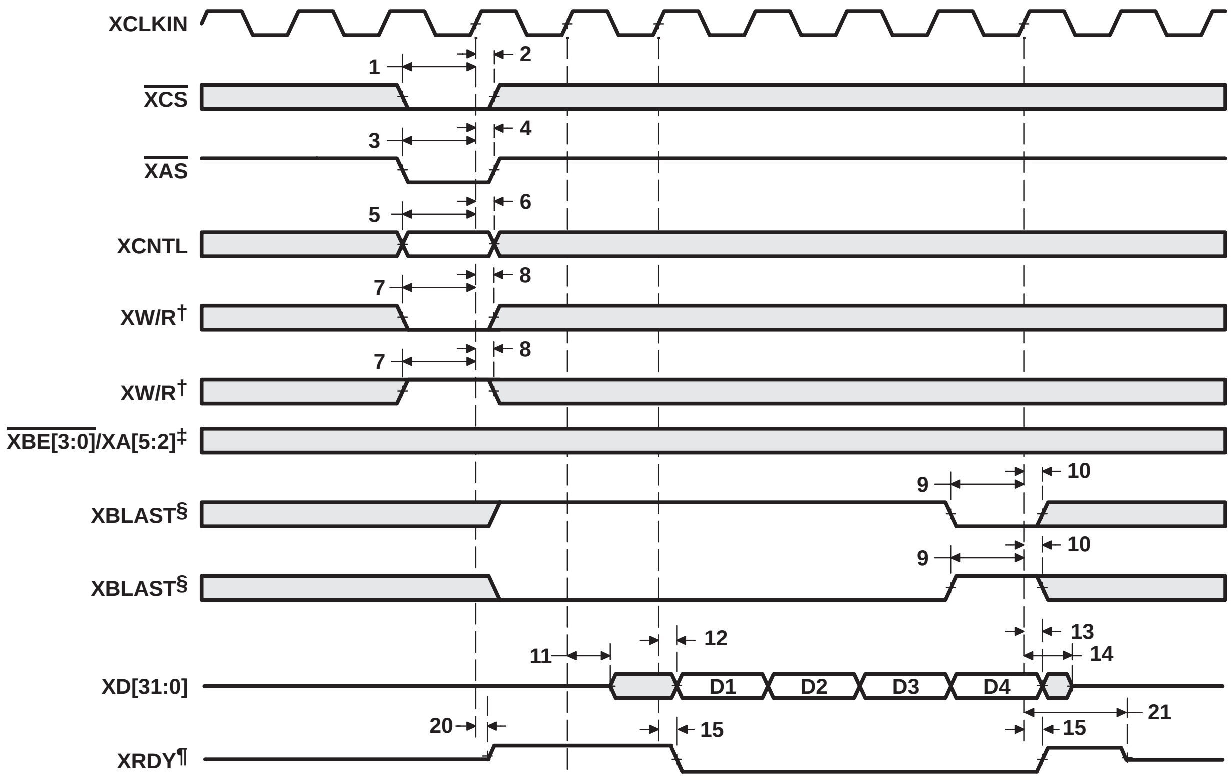
### switching characteristics with external device as bus master¶ (see Figure 31 and Figure 32)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 11 | $t_{d(XCKIH-XDLZ)}$ | Delay time, XCLKIN high to XDx low impedance | 5 | | ns |
| 12 | $t_{d(XCKIH-XDV)}$ | Delay time, XCLKIN high to XDx valid | | 15.5 | ns |
| 13 | $t_{d(XCKIH-XDIV)}$ | Delay time, XCLKIN high to XDx invalid | 5 | | ns |
| 14 | $t_{d(XCKIH-XDHZ)}$ | Delay time, XCLKIN high to XDx high impedance | | 18 | ns |
| 15 | $t_{d(XCKIH-XRY)}$ | Delay time, XCLKIN high to XRDY valid# | 5 | 15.5 | ns |
| 20 | $t_{d(XCKIH-XRYLZ)}$ | Delay time, XCLKIN high to XRDY low impedance | 5 | 15.5 | ns |
| 21 | $t_{d(XCKIH-XRYHZ)}$ | Delay time, XCLKIN high to XRDY high impedance# | 2P + 5 | 3P + 15.5 | ns |

¶ P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
# XRDY operates as active-low ready input/output during host-port accesses.

ADVANCE INFORMATION

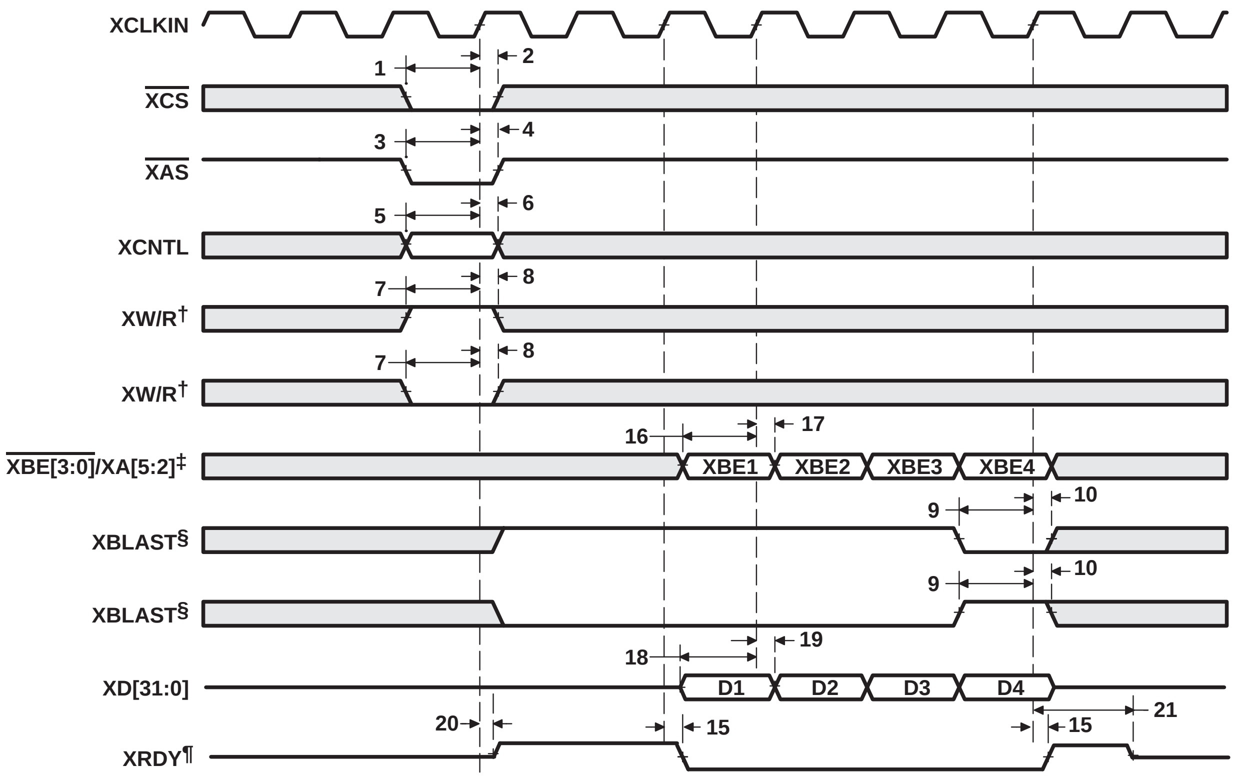
## EXPANSION BUS SYNCHRONOUS HOST PORT TIMING (CONTINUED)




† XW/R input/output polarity selected at boot
‡ XBE[3:0]/XA[5:2] operates as byte enables XBE[3:0] during host-port accesses.
§ XBLAST input polarity selected at boot
¶ XRDY operates as active-low ready input/output during host-port accesses.

**Figure 31. External Host as Bus Master—Read**

## EXPANSION BUS SYNCHRONOUS HOST PORT TIMING (CONTINUED)




† XW/R input/output polarity selected at boot
‡ XBE[3:0]/XA[5:2] operates as byte enables XBE[3:0] during host-port accesses.
§ XBLAST input polarity selected at boot
¶ XRDY operates as active-low ready input/output during host-port accesses.

**Figure 32. External Host as Bus Master—Write**

ADVANCE INFORMATION

## EXPANSION BUS SYNCHRONOUS HOST PORT TIMING (CONTINUED)

### timing requirements with ’C6202 as bus master (see Figure 33, Figure 34, and Figure 35)

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 9 | $t_{su(XDV-XCKIH)}$ | Setup time, XDx valid before XCLKIN high | 4 | | ns |
| 10 | $t_{h(XCKIH-XDV)}$ | Hold time, XDx valid after XCLKIN high | 2.3 | | ns |
| 11 | $t_{su(XRY-XCKIH)}$ | Setup time, XRDY valid before XCLKIN high† | 4 | | ns |
| 12 | $t_{h(XCKIH-XRY)}$ | Hold time, XRDY valid after XCLKIN high† | 2.3 | | ns |
| 14 | $t_{su(XBFF-XCKIH)}$ | Setup time, XBOFF valid before XCLKIN high | 4 | | ns |
| 15 | $t_{h(XCKIH-XBFF)}$ | Hold time, XBOFF valid after XCLKIN high | 2.3 | | ns |

† XRDY operates as active-low ready input/output during host-port accesses.

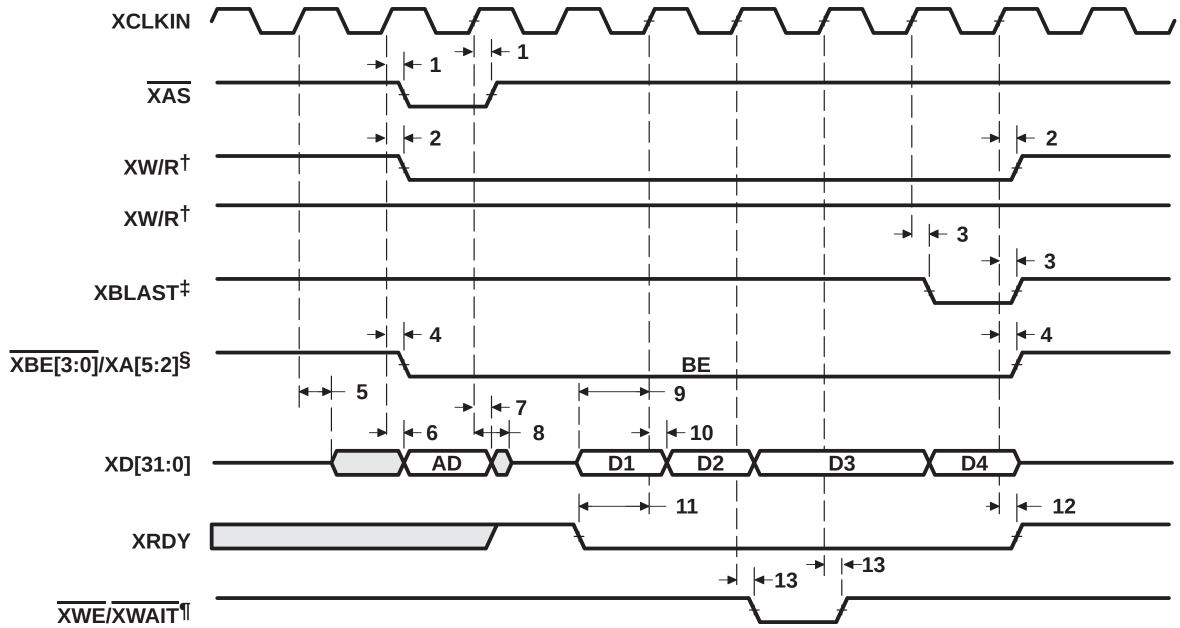
### switching characteristics with ’C6202 as bus master (see Figure 33, Figure 34, and Figure 35)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{d(XCKIH-XASV)}$ | Delay time, XCLKIN high to $\overline{\text{XAS}}$ valid | 5 | 15.5 | ns |
| 2 | $t_{d(XCKIH-XWRV)}$ | Delay time, XCLKIN high to XW/R valid‡ | 5 | 15.5 | ns |
| 3 | $t_{d(XCKIH-XBLTV)}$ | Delay time, XCLKIN high to XBLAST valid§ | 5 | 15.5 | ns |
| 4 | $t_{d(XCKIH-XBEV)}$ | Delay time, XCLKIN high to $\overline{\text{XBE[3:0]}}$/XA[5:2] valid¶ | 5 | 15.5 | ns |
| 5 | $t_{d(XCKIH-XDLZ)}$ | Delay time, XCLKIN high to XDx low impedance | 5 | | ns |
| 6 | $t_{d(XCKIH-XDV)}$ | Delay time, XCLKIN high to XDx valid | | 15.5 | ns |
| 7 | $t_{d(XCKIH-XDIV)}$ | Delay time, XCLKIN high to XDx invalid | 5 | | ns |
| 8 | $t_{d(XCKIH-XDHZ)}$ | Delay time, XCLKIN high to XDx high impedance | | 18 | ns |
| 13 | $t_{d(XCKIH-XWTV)}$ | Delay time, XCLKIN high to $\overline{\text{XWE}}/\overline{\text{XWAIT}}$ valid# | 5 | 15.5 | ns |

‡ XW/R input/output polarity selected at boot.
§ XBLAST output polarity is always active low.
¶ XBE[3:0]/XA[5:2] operates as byte enables XBE[3:0] during host-port accesses.
# XWE/XWAIT operates as XWAIT output signal during host-port accesses.

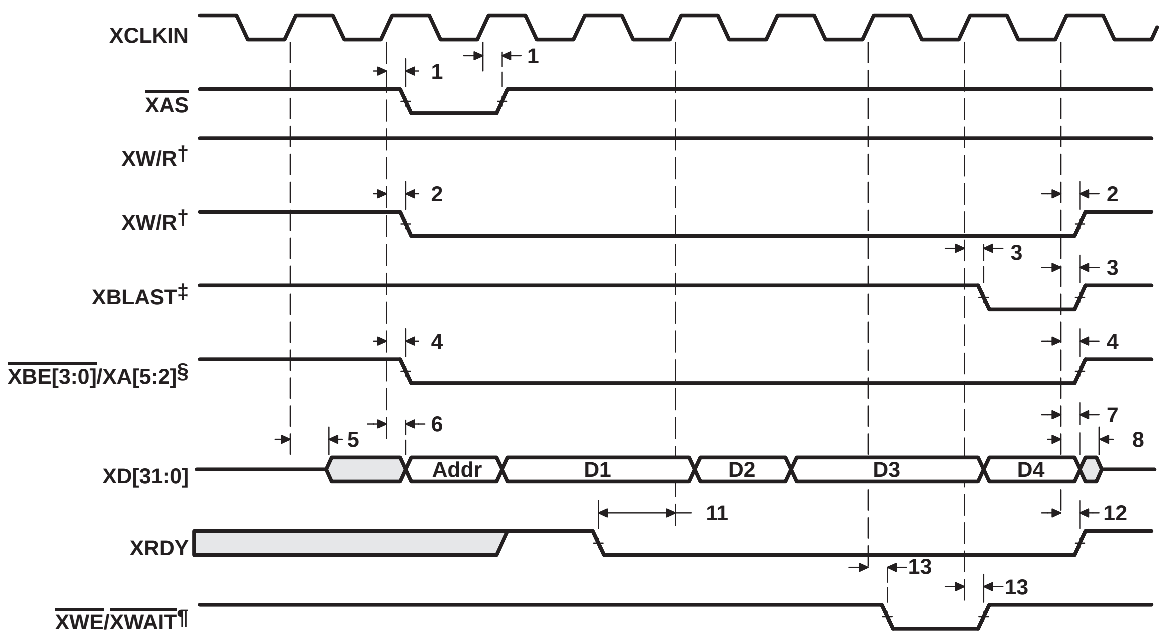
ADVANCE INFORMATION

## EXPANSION BUS SYNCHRONOUS HOST PORT TIMING (CONTINUED)




† XW/R input/output polarity selected at boot
‡ XBLAST output polarity is always active low.
§ XBE[3:0]/XA[5:2] operates as byte enables XBE[3:0] during host-port accesses.
¶ XWE/XWAIT operates as XWAIT output signal during host-port accesses.

**Figure 33. ’C6202 as Bus Master—Read**

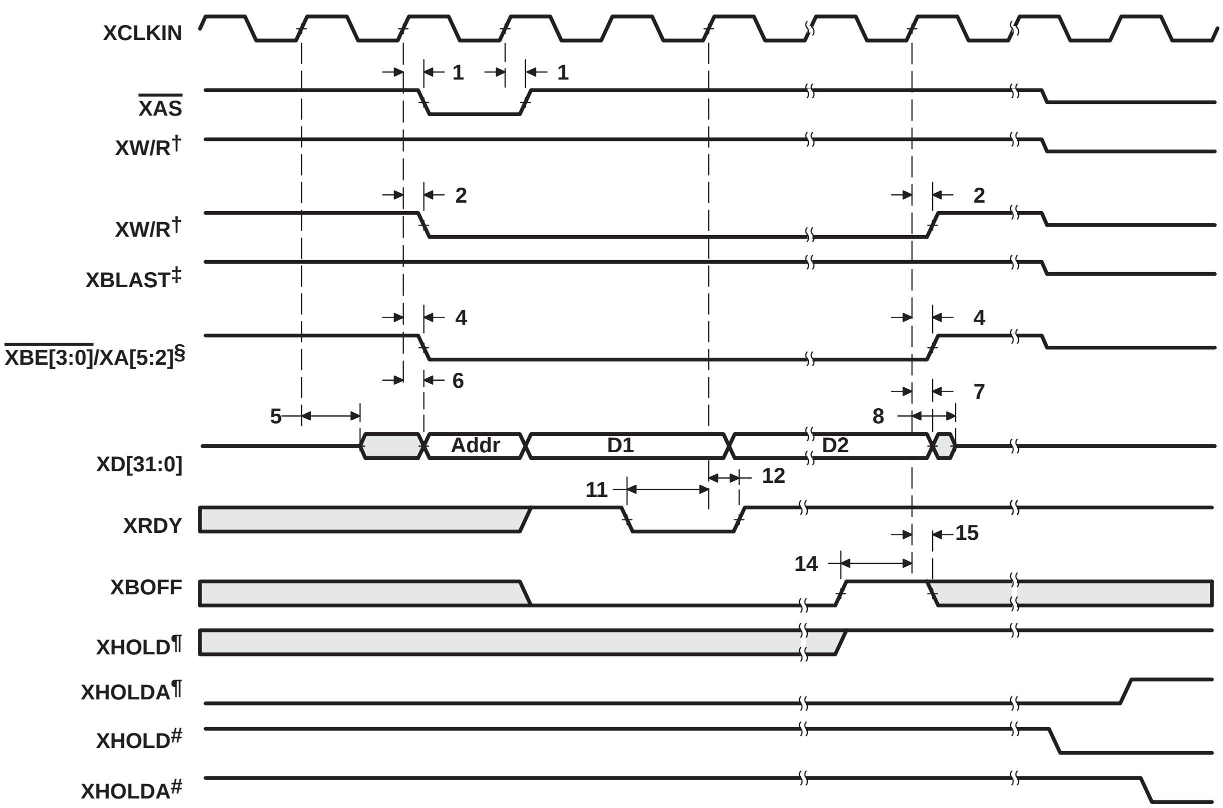



† XW/R input/output polarity selected at boot
‡ XBLAST output polarity is always active low.
§ XBE[3:0]/XA[5:2] operates as byte enables XBE[3:0] during host-port accesses.
¶ XWE/XWAIT operates as XWAIT output signal during host-port accesses.

**Figure 34. ’C6202 as Bus Master—Write**

ADVANCE INFORMATION

## EXPANSION BUS SYNCHRONOUS HOST PORT TIMING (CONTINUED)




† XW/R input/output polarity selected at boot
‡ XBLAST output polarity is always active low.
§ XBE[3:0]/XA[5:2] operates as byte enables XBE[3:0] during host-port accesses.
¶ Internal arbiter enabled
# External arbiter enabled
‖ This diagram illustrates XBOFF timing. Bus arbitration timing is shown in Figure 38 and Figure 39.

**Figure 35. ’C6202 as Bus Master—BOFF Operation‖**

## EXPANSION BUS ASYNCHRONOUS HOST PORT TIMING

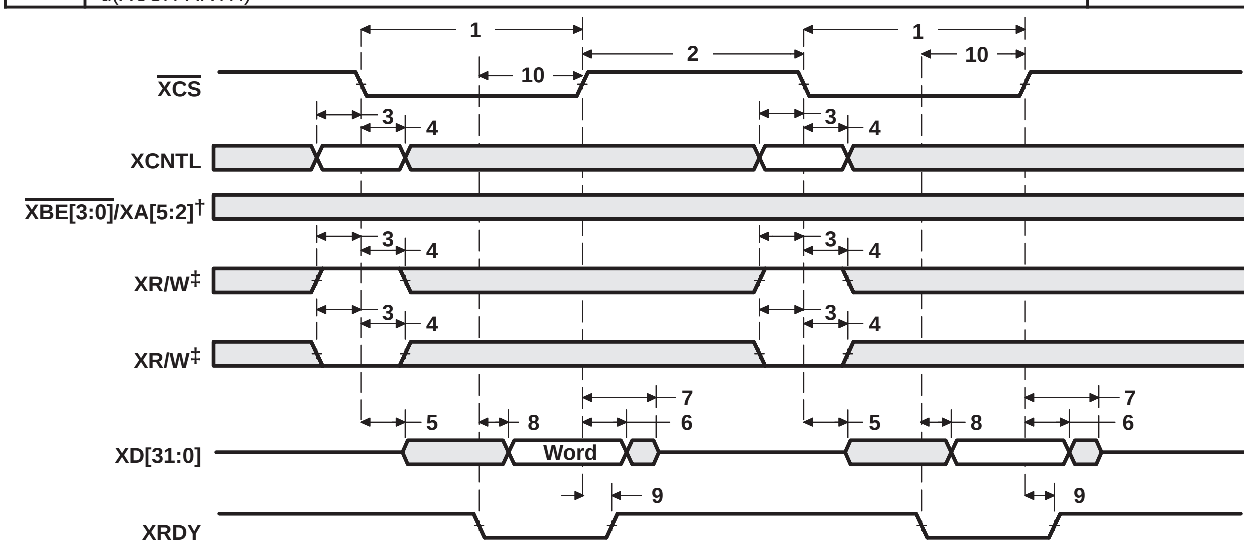
### timing requirements with external device as asynchronous bus master† (see Figure 36 and Figure 37)

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{w(XCSL)}$ | Pulse duration, $\overline{\text{XCS}}$ low | 4P | | ns |
| 2 | $t_{w(XCSH)}$ | Pulse duration, $\overline{\text{XCS}}$ high | 4P | | ns |
| 3 | $t_{su(XSEL-XCSL)}$ | Setup time, expansion bus select signals‡ valid before $\overline{\text{XCS}}$ low | 2 | | ns |
| 4 | $t_{h(XCSL-XSEL)}$ | Hold time, expansion bus select signals‡ valid after $\overline{\text{XCS}}$ low | 2 | | ns |
| 10 | $t_{h(XRYL-XCSL)}$ | Hold time, $\overline{\text{XCS}}$ low after XRDY low | P | | ns |
| 11 | $t_{su(XBEV-XCSH)}$ | Setup time, $\overline{\text{XBE[3:0]}}$/XA[5:2] valid before $\overline{\text{XCS}}$ high§ | 2 | | ns |
| 12 | $t_{h(XCSH-XBEV)}$ | Hold time, $\overline{\text{XBE[3:0]}}$/XA[5:2] valid after $\overline{\text{XCS}}$ high§ | 2 | | ns |
| 13 | $t_{su(XDV-XCSH)}$ | Setup time, XDx valid before $\overline{\text{XCS}}$ high | 2 | | ns |
| 14 | $t_{h(XCSH-XDV)}$ | Hold time, XDx valid after $\overline{\text{XCS}}$ high | 2 | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ Expansion bus select signals include XCNTL and XR/W.
§ $\overline{\text{XBE[3:0]}}$/XA[5:2] operates as byte enables $\overline{\text{XBE[3:0]}}$ during host-port accesses.

### switching characteristics with external device as asynchronous bus master (see Figure 36 and Figure 37)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 5 | $t_{d(XCSL-XDLZ)}$ | Delay time, $\overline{\text{XCS}}$ low to XDx low impedance | 0 | | ns |
| 6 | $t_{d(XCSH-XDIV)}$ | Delay time, $\overline{\text{XCS}}$ high to XDx invalid | 0 | 12 | ns |
| 7 | $t_{d(XCSH-XDHZ)}$ | Delay time, $\overline{\text{XCS}}$ high to XDx high impedance | | 12 | ns |
| 8 | $t_{d(XRYL-XDV)}$ | Delay time, XRDY low to XDx valid | 0 | 4 | ns |
| 9 | $t_{d(XCSH-XRYH)}$ | Delay time, $\overline{\text{XCS}}$ high to XRDY high | 0 | 12 | ns |

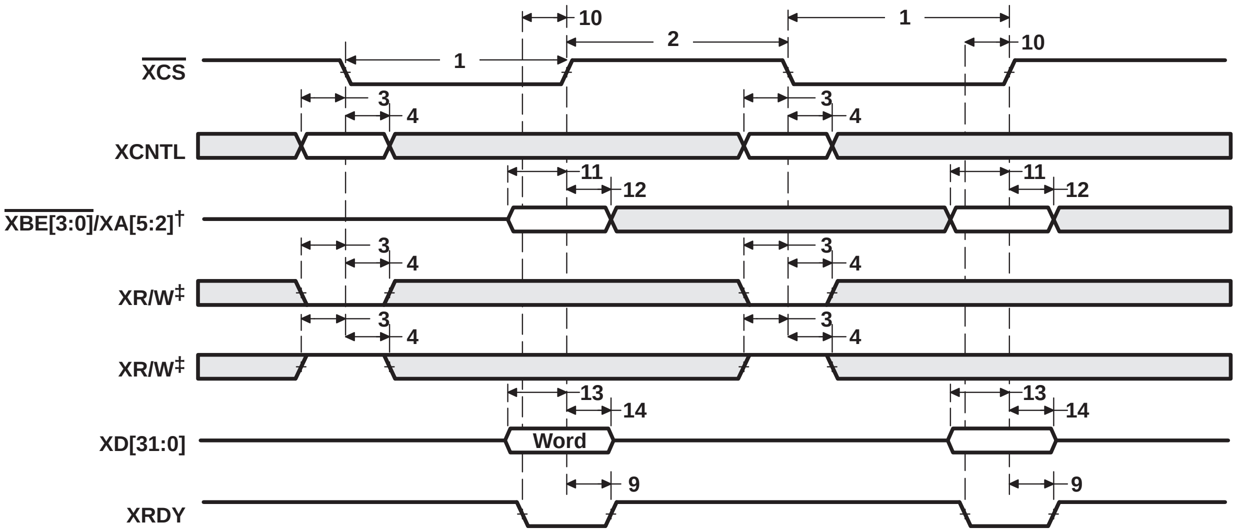



† $\overline{\text{XBE[3:0]}}$/XA[5:2] operates as byte enables $\overline{\text{XBE[3:0]}}$ during host-port accesses.
‡ XW/R input/output polarity selected at boot

**Figure 36. External Device as Asynchronous Master—Read**

ADVANCE INFORMATION

## EXPANSION BUS ASYNCHRONOUS HOST PORT TIMING (CONTINUED)




† XBE[3:0]/XA[5:2] operates as byte enables XBE[3:0] during host-port accesses.
‡ XW/R input/output polarity selected at boot

**Figure 37. External Device as Asynchronous Master—Write**

ADVANCE INFORMATION

## XHOLD/XHOLDA TIMING

### timing requirements for expansion bus arbitration (internal arbiter enabled)† (see Figure 38)

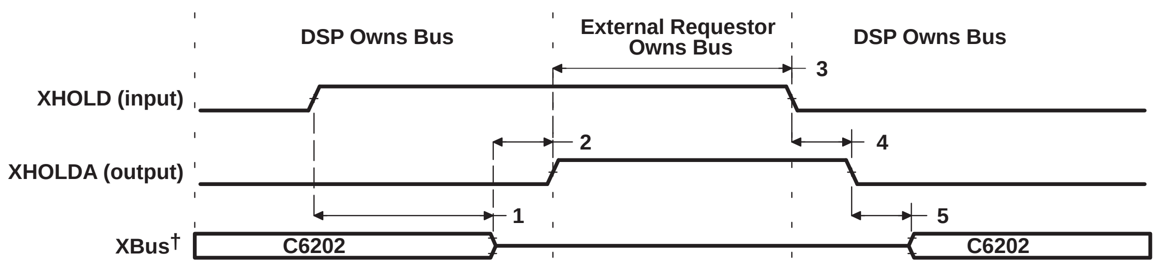
| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 3 | $t_{oh(XHDAH-XHDH)}$ | Output hold time, XHOLD high after XHOLDA high | P | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

### switching characteristics for expansion bus arbitration (internal arbiter enabled)†‡ (see Figure 38)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{R(XHDH-XBHZ)}$ | Response time, XHOLD high to XBus high impedance | 4P | § | ns |
| 2 | $t_{d(XBHZ-XHDAH)}$ | Delay time, XBus high impedance to XHOLDA high | 0 | 2P | ns |
| 4 | $t_{R(XHDL-XHDAL)}$ | Response time, XHOLD low to XHOLDA low | 4P | | ns |
| 5 | $t_{d(XHDAL-XBLZ)}$ | Delay time, XHOLDA low to XBus low impedance | 0 | 2P | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ XBus consists of $\overline{XBE[3:0]}$/XA[5:2], $\overline{XAS}$, XW/R, and XBLAST.
§ All pending XBus transactions are allowed to complete before XHOLDA is asserted.




† XBus consists of $\overline{XBE[3:0]}$/XA[5:2], $\overline{XAS}$, XW/R, and XBLAST.

**Figure 38. Expansion Bus Arbitration—Internal Arbiter Enabled**

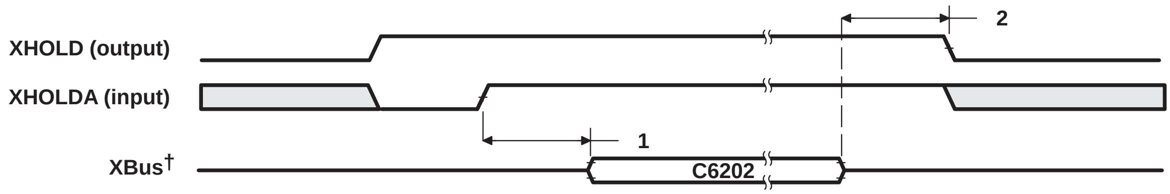
ADVANCE INFORMATION

## XHOLD/XHOLDA TIMING (CONTINUED)

### switching characteristics for expansion bus arbitration (internal arbiter disabled)† (see Figure 39)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{d(XHDAH\text{-}XBLZ)}$ | Delay time, XHOLDA high to XBus low impedance‡ | 2P | 2P + 10 | ns |
| 2 | $t_{d(XBHZ\text{-}XHDL)}$ | Delay time, XBus high impedance to XHOLD low‡ | 0 | 2P | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ XBus consists of $\overline{\text{XBE[3:0]}}$/XA[5:2], $\overline{\text{XAS}}$, XW/R, and XBLAST.




† XBus consists of $\overline{\text{XBE[3:0]}}$/XA[5:2], $\overline{\text{XAS}}$, XW/R, and XBLAST.

**Figure 39. Expansion Bus Arbitration—Internal Arbiter Disabled**

ADVANCE INFORMATION

## MULTICHANNEL BUFFERED SERIAL PORT TIMING

### timing requirements for McBSP†‡ (see Figure 40)

| NO. | | | | ’C6202-200 ’C6202-233 ’C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|---|
| 2 | $t_{c(CKRX)}$ | Cycle time, CLKR/X | CLKR/X ext | 2P | | ns |
| 3 | $t_{w(CKRX)}$ | Pulse duration, CLKR/X high or CLKR/X low | CLKR/X ext | P – 1 | | ns |
| 5 | $t_{su(FRH-CKRL)}$ | Setup time, external FSR high before CLKR low | CLKR int | 9 | | ns |
| | | | CLKR ext | 1 | | |
| 6 | $t_{h(CKRL-FRH)}$ | Hold time, external FSR high after CLKR low | CLKR int | 6 | | ns |
| | | | CLKR ext | 3 | | |
| 7 | $t_{su(DRV-CKRL)}$ | Setup time, DR valid before CLKR low | CLKR int | 8 | | ns |
| | | | CLKR ext | 0 | | |
| 8 | $t_{h(CKRL-DRV)}$ | Hold time, DR valid after CLKR low | CLKR int | 3 | | ns |
| | | | CLKR ext | 3 | | |
| 10 | $t_{su(FXH-CKXL)}$ | Setup time, external FSX high before CLKX low | CLKX int | 9 | | ns |
| | | | CLKX ext | 1 | | |
| 11 | $t_{h(CKXL-FXH)}$ | Hold time, external FSX high after CLKX low | CLKX int | 6 | | ns |
| | | | CLKX ext | 3 | | |

† CLKRP = CLKXP = FSRP = FSXP = 0. If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
‡ P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

ADVANCE INFORMATION

ADVANCE INFORMATION

## MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

### switching characteristics for McBSP†‡ (see Figure 40)

| NO. | PARAMETER | | | ’C6202-200 ’C6202-233 ’C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|---|
| 1 | $t_{d(CKSH-CKRXH)}$ | Delay time, CLKS high to CLKR/X high for internal CLKR/X generated from CLKS input | | 4 | 10 | ns |
| 2 | $t_{c(CKRX)}$ | Cycle time, CLKR/X | CLKR/X int | 2P§ | | ns |
| 3 | $t_{w(CKRX)}$ | Pulse duration, CLKR/X high or CLKR/X low | CLKR/X int | C – 1¶ | C + 1¶ | ns |
| 4 | $t_{d(CKRH-FRV)}$ | Delay time, CLKR high to internal FSR valid | CLKR int | –2 | 3 | ns |
| 9 | $t_{d(CKXH-FXV)}$ | Delay time, CLKX high to internal FSX valid | CLKX int | –2 | 3 | ns |
| | | | CLKX ext | 3 | 9 | |
| 12 | $t_{dis(CKXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX high | CLKX int | –1 | 4 | ns |
| | | | CLKX ext | 3 | 9 | |
| 13 | $t_{d(CKXH-DXV)}$ | Delay time, CLKX high to DX valid | CLKX int | –1 | 4 | ns |
| | | | CLKX ext | 3 | 9 | |
| 14 | $t_{d(FXH-DXV)}$ | Delay time, FSX high to DX valid<br>ONLY applies when in data delay 0 (XDATDLY = 00b) mode. | FSX int | –1 | 3 | ns |
| | | | FSX ext | 3 | 9 | |

† CLKRP = CLKXP = FSRP = FSXP = 0. If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
‡ Minimum delay times also represent minimum output hold times.
§ P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
¶ C = H or L
S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero

## MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

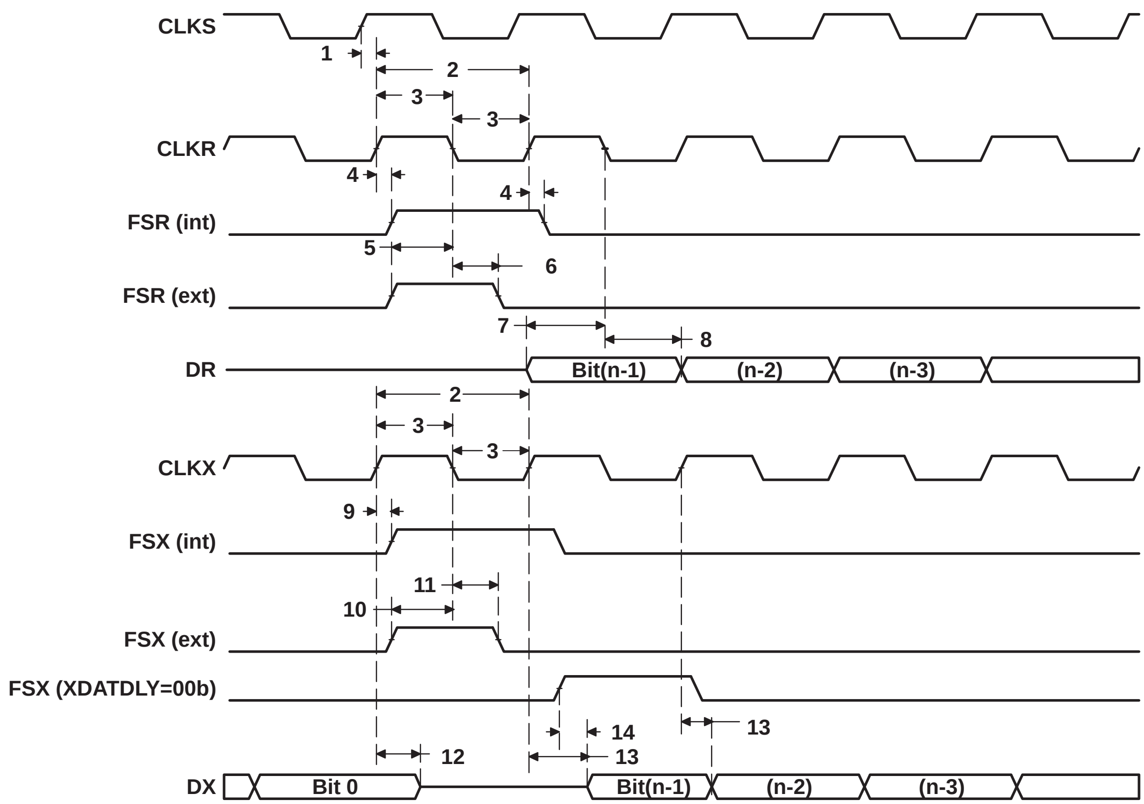



Figure 40. McBSP Timings

ADVANCE INFORMATION

## MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

### timing requirements for FSR when GSYNC = 1 (see Figure 41)

| NO. | | | ’C6202-200 ’C6202-233 ’C6202-250 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{su(FRH-CKSH)}$ | Setup time, FSR high before CLKS high | 4 | | ns |
| 2 | $t_{h(CKSH-FRH)}$ | Hold time, FSR high after CLKS high | 4 | | ns |

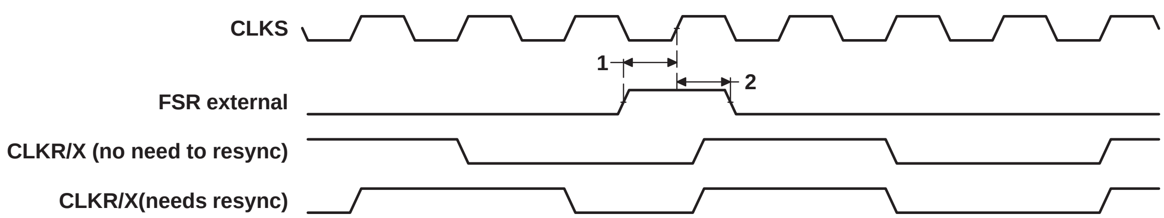



Figure 41. FSR Timing When GSYNC = 1

ADVANCE INFORMATION

## MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

### timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 0[†‡] (see Figure 42)

| NO. | | | ’C6202-200 ’C6202-233 ’C6202-250 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 4 | $t_{su(DRV-CKXL)}$ | Setup time, DR valid before CLKX low | 12 | | 2 – 3P | | ns |
| 5 | $t_{h(CKXL-DRV)}$ | Hold time, DR valid after CLKX low | 4 | | 5 + 6P | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

### switching characteristics for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 0[†‡] (see Figure 42)

| NO. | PARAMETER | | ’C6202-200 ’C6202-233 ’C6202-250 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER[§] | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXL-FXL)}$ | Hold time, FSX low after CLKX low[¶] | T – 2 | T + 3 | | | ns |
| 2 | $t_{d(FXL-CKXH)}$ | Delay time, FSX low to CLKX high[#] | L – 2 | L + 3 | | | ns |
| 3 | $t_{d(CKXH-DXV)}$ | Delay time, CLKX high to DX valid | –2 | 4 | 3P + 4 | 5P + 17 | ns |
| 6 | $t_{dis(CKXL-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX low | L – 2 | L + 3 | | | ns |
| 7 | $t_{dis(FXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from FSX high | | | P + 3 | 3P + 17 | ns |
| 8 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | | | 2P + 2 | 4P + 17 | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

ADVANCE INFORMATION

## MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

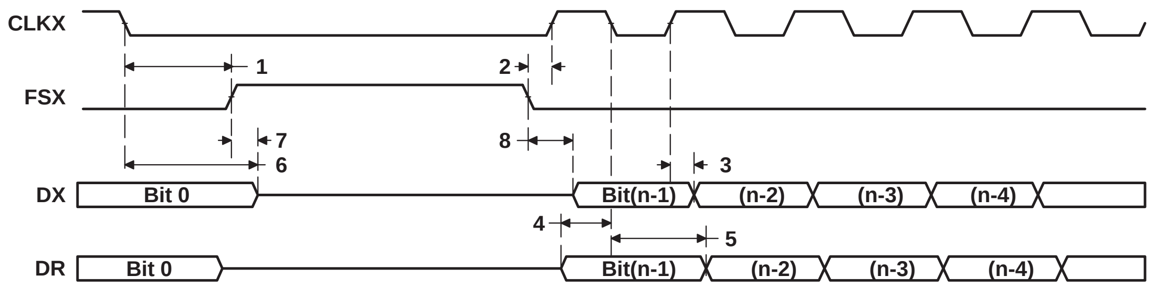



Figure 42. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

ADVANCE INFORMATION

## MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

### timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 43)

| NO. | | | ’C6202-200 ’C6202-233 ’C6202-250 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 4 | $t_{su(DRV-CKXH)}$ | Setup time, DR valid before CLKX high | 12 | | 2 – 3P | | ns |
| 5 | $t_{h(CKXH-DRV)}$ | Hold time, DR valid after CLKX high | 4 | | 5 + 6P | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

### switching characteristics for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 43)

| NO. | PARAMETER | | ’C6202-200 ’C6202-233 ’C6202-250 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER§ | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXL-FXL)}$ | Hold time, FSX low after CLKX low¶ | L – 2 | L + 3 | | | ns |
| 2 | $t_{d(FXL-CKXH)}$ | Delay time, FSX low to CLKX high# | T – 2 | T + 3 | | | ns |
| 3 | $t_{d(CKXL-DXV)}$ | Delay time, CLKX low to DX valid | –2 | 4 | 3P + 4 | 5P + 17 | ns |
| 6 | $t_{dis(CKXL-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX low | –2 | 4 | 3P + 3 | 5P + 17 | ns |
| 7 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | H – 2 | H + 4 | 2P + 2 | 4P + 17 | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

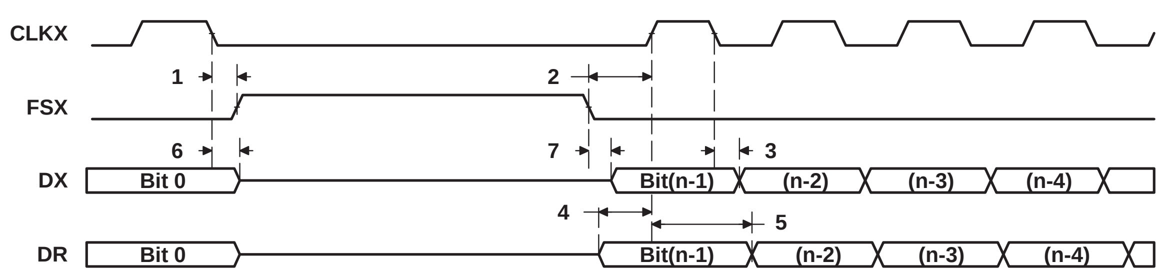



Figure 43. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

ADVANCE INFORMATION

## MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

### timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 44)

| NO. | | | ’C6202-200 ’C6202-233 ’C6202-250 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 4 | $t_{su(DRV-CKXH)}$ | Setup time, DR valid before CLKX high | 12 | | 2 – 3P | | ns |
| 5 | $t_{h(CKXH-DRV)}$ | Hold time, DR valid after CLKX high | 4 | | 5 + 6P | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

### switching characteristics for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 44)

| NO. | PARAMETER | | ’C6202-200 ’C6202-233 ’C6202-250 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER§ | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXH-FXL)}$ | Hold time, FSX low after CLKX high¶ | T – 2 | T + 3 | | | ns |
| 2 | $t_{d(FXL-CKXL)}$ | Delay time, FSX low to CLKX low# | H – 2 | H + 3 | | | ns |
| 3 | $t_{d(CKXL-DXV)}$ | Delay time, CLKX low to DX valid | –2 | 4 | 3P + 4 | 5P + 17 | ns |
| 6 | $t_{dis(CKXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX high | H – 2 | H + 3 | | | ns |
| 7 | $t_{dis(FXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from FSX high | | | P + 3 | 3P + 17 | ns |
| 8 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | | | 2P + 2 | 4P + 17 | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

ADVANCE INFORMATION

## MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

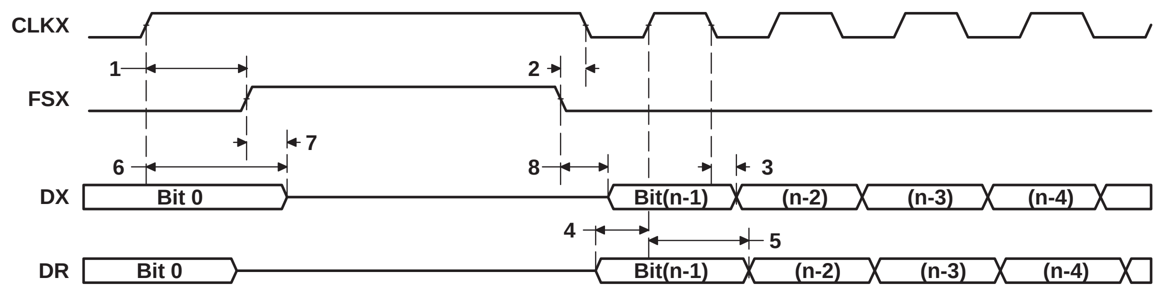



**Figure 44. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1**

ADVANCE INFORMATION

## MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

### timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 45)

| NO. | | | ’C6202-200 ’C6202-233 ’C6202-250 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 4 | $t_{su(DRV-CKXL)}$ | Setup time, DR valid before CLKX low | 12 | | 2 – 3P | | ns |
| 5 | $t_{h(CKXL-DRV)}$ | Hold time, DR valid after CLKX low | 4 | | 5 + 6P | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

### switching characteristics for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 45)

| NO. | PARAMETER | | ’C6202-200 ’C6202-233 ’C6202-250 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER§ | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXH-FXL)}$ | Hold time, FSX low after CLKX high¶ | H – 2 | H + 3 | | | ns |
| 2 | $t_{d(FXL-CKXL)}$ | Delay time, FSX low to CLKX low# | T – 2 | T + 1 | | | ns |
| 3 | $t_{d(CKXH-DXV)}$ | Delay time, CLKX high to DX valid | –2 | 4 | 3P + 4 | 5P + 17 | ns |
| 6 | $t_{dis(CKXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX high | –2 | 4 | 3P + 3 | 5P + 17 | ns |
| 7 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | L – 2 | L + 4 | 2P + 2 | 4P + 17 | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
 = sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
 = (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
 = (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

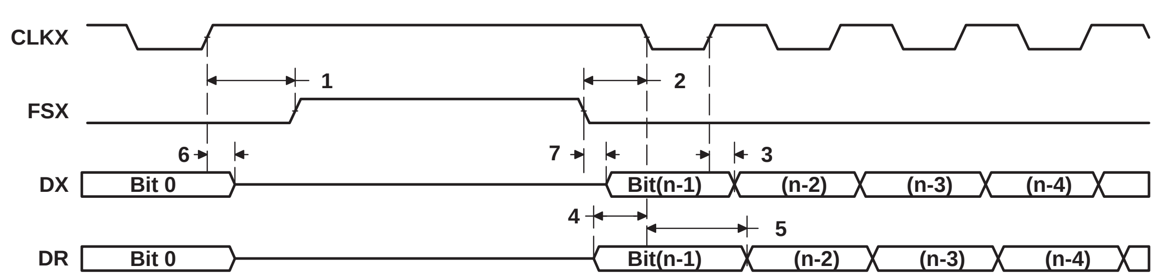



Figure 45. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

ADVANCE INFORMATION

## DMAC, TIMER, POWER-DOWN TIMING

### switching characteristics for DMAC outputs† (see Figure 46)

| NO. | PARAMETER | | 'C6202-200 'C6202-233 'C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{w(DMACH)}$ | Pulse duration, DMAC high | 2P–3 | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.




**Figure 46. DMAC Timing**

### timing requirements for timer inputs† (see Figure 47)

| NO. | | | 'C6202-200 'C6202-233 'C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{w(TINPH)}$ | Pulse duration, TINP high | 2P | | ns |
| 2 | $t_{w(TINPL)}$ | Pulse duration, TINP low | 2P | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

### switching characteristics for timer outputs† (see Figure 47)

| NO. | PARAMETER | | 'C6202-200 'C6202-233 'C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 3 | $t_{w(TOUTH)}$ | Pulse duration, TOUT high | 2P–3 | | ns |
| 4 | $t_{w(TOUTL)}$ | Pulse duration, TOUT low | 2P–3 | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

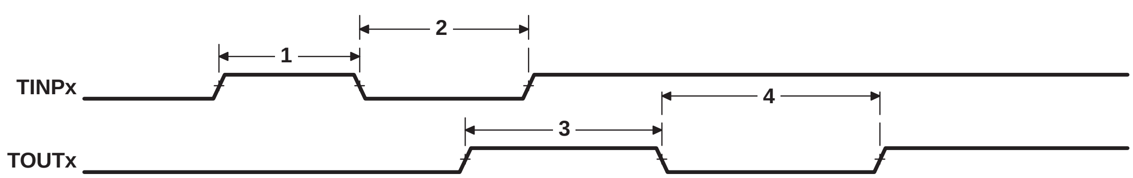



**Figure 47. Timer Timing**

ADVANCE INFORMATION

## DMAC, TIMER, POWER-DOWN TIMING (CONTINUED)

### switching characteristics for power-down outputs† (see Figure 48)

| NO. | PARAMETER | | 'C6202-200 'C6202-233 'C6202-250 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(PDH)}$ | Pulse duration, PD high | 10P | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

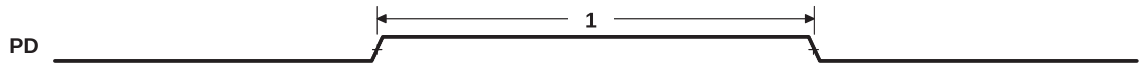



Figure 48. Power-Down Timing

ADVANCE INFORMATION

## JTAG TEST-PORT TIMING

### timing requirements for JTAG test port (see Figure 49)

| NO. | | | ’C6202-200 ’C6202-233 ’C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{c(TCK)}$ | Cycle time, TCK | | 50 | ns |
| 3 | $t_{su(TDIV-TCKH)}$ | Setup time, TDI/TMS/$\overline{\text{TRST}}$ valid before TCK high | 10 | | ns |
| 4 | $t_{h(TCKH-TDIV)}$ | Hold time, TDI/TMS/$\overline{\text{TRST}}$ valid after TCK high | 5 | | ns |

### switching characteristics for JTAG test port (see Figure 49)

| NO. | PARAMETER | | ’C6202-200 ’C6202-233 ’C6202-250 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 2 | $t_{d(TCKL-TDOV)}$ | Delay time, TCK low to TDO valid | 0 | 15 | ns |

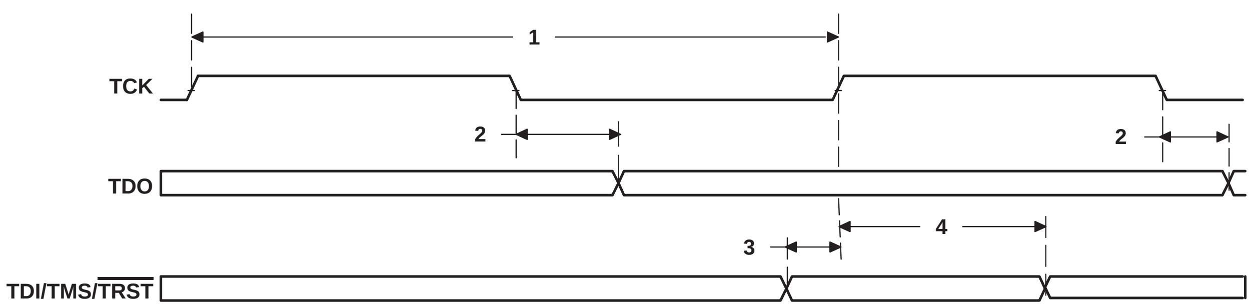


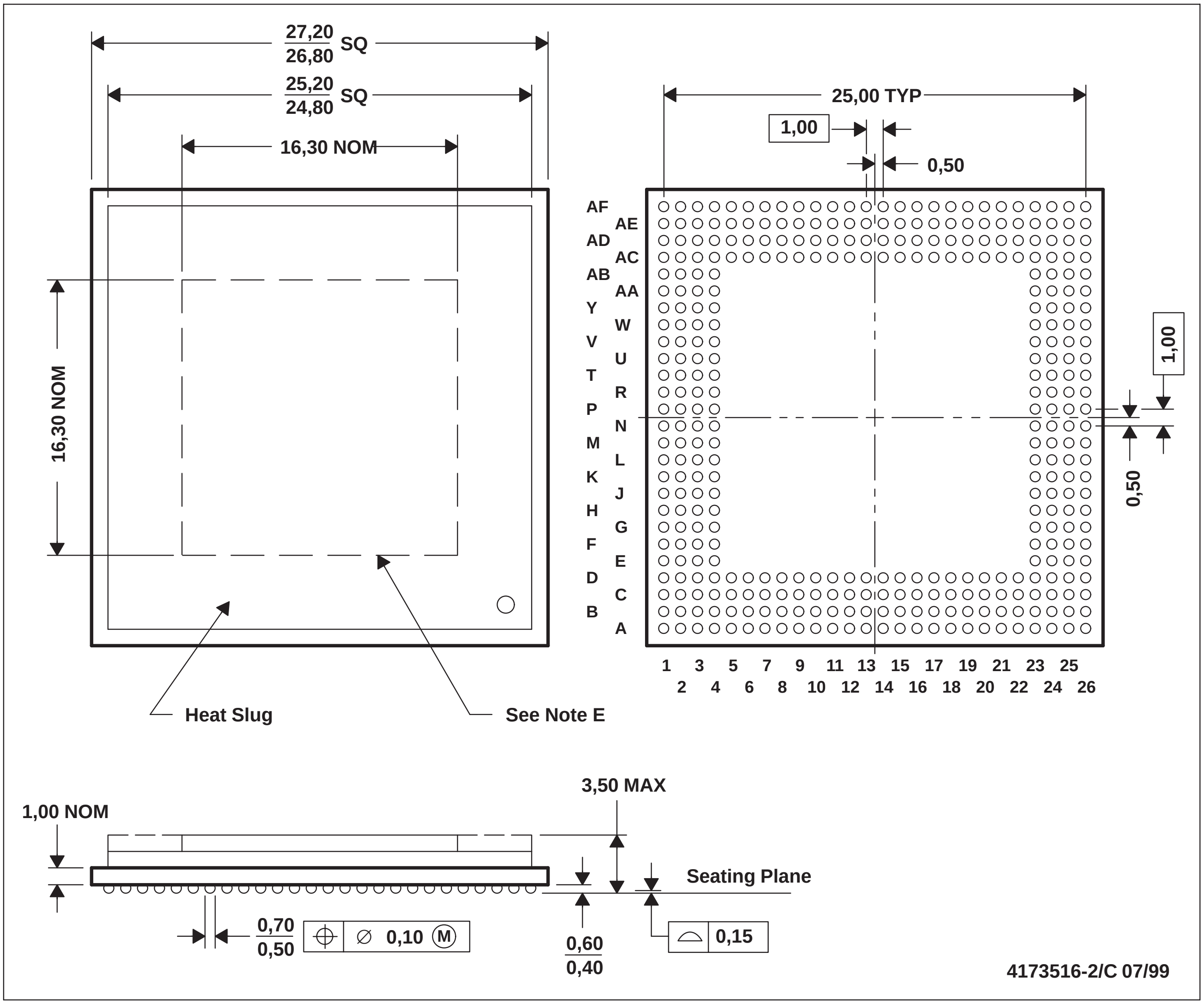

Figure 49. JTAG Test-Port Timing

ADVANCE INFORMATION

ADVANCE INFORMATION

## MECHANICAL DATA

**GJL (S-PBGA-N352)** **PLASTIC BALL GRID ARRAY**




NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Thermally enhanced plastic package with heat slug (HSL).
D. Flip chip application only
E. Possible protrusion in this area, but within 3,50 max package height specification
F. Falls within JEDEC MO-151/AAL-1

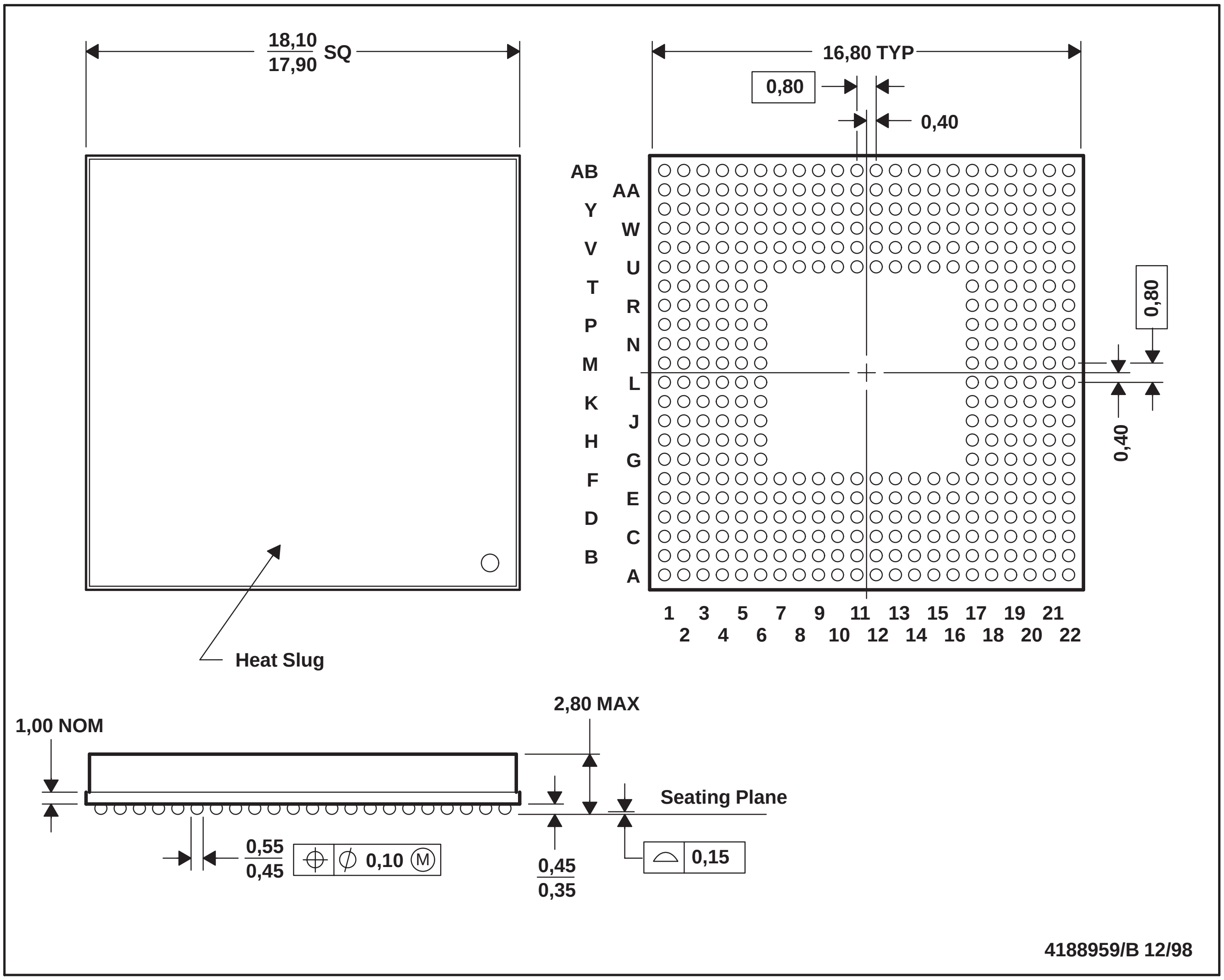
## thermal resistance characteristics (S-PBGA package)

| NO | | | °C/W | Air Flow LFPM† |
|---|---|---|---|---|
| 1 | $R\Theta_{JC}$ | Junction-to-case | 0.47 | N/A |
| 2 | $R\Theta_{JA}$ | Junction-to-free air | 14.2 | 0 |
| 3 | $R\Theta_{JA}$ | Junction-to-free air | 12.3 | 100 |
| 4 | $R\Theta_{JA}$ | Junction-to-free air | 10.2 | 250 |
| 5 | $R\Theta_{JA}$ | Junction-to-free air | 8.6 | 500 |

† LFPM = Linear Feet Per Minute

## MECHANICAL DATA

**GLS (S-PBGA-N384)** **PLASTIC BALL GRID ARRAY**






NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Thermally enhanced plastic package with heat slug (HSL)
D. Flip chip application only

## thermal resistance characteristics (S-PBGA package)

| NO | | | °C/W | Air Flow LFPM† |
|---|---|---|---|---|
| 1 | $R\Theta_{JC}$ | Junction-to-case | 0.85 | N/A |
| 2 | $R\Theta_{JA}$ | Junction-to-free air | 21.6 | 0 |
| 3 | $R\Theta_{JA}$ | Junction-to-free air | 17.9 | 100 |
| 4 | $R\Theta_{JA}$ | Junction-to-free air | 14.2 | 250 |
| 5 | $R\Theta_{JA}$ | Junction-to-free air | 11.8 | 500 |

† LFPM = Linear Feet Per Minute